DTIC FILE COPY

ARO 24567.2-EL

(2)

JC-2044-A

AD-A197 738

# THE DIVERSITY ECCM PERFORMANCE OF FREQUENCY-HOPPING CPFSK IN PARTIAL-BAND NOISE JAMMING

FINAL REPORT

MAY 25, 1988

PREPARED FOR

U. S. ARMY RESEARCH OFFICE

CONTRACT

DAAL03-87-C-0006

The views, opinions, and/or findings contained in this report are those of the authors and *should* not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

DTIC
ELECTE
JUL 20 1988
S D
H

J. S. LEE ASSOCIATES, INC.
2001 JEFFERSON DAVIS HIGHWAY, SUITE 601
ARLINGTON, VIRGINIA 22202

Approved for public release; distribution unlimited.

A1 723

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

**1a. REPORT SECURITY CLASSIFICATION:** Unclassified

**1b. RESTRICTIVE MARKINGS:**

**2a. SECURITY CLASSIFICATION AUTHORITY:**

**2b. DECLASSIFICATION / DOWNGRADING SCHEDULE:**

**3. DISTRIBUTION / AVAILABILITY OF REPORT:** Approved for public release; distribution unlimited.

**4. PERFORMING ORGANIZATION REPORT NUMBER(S):** JC-2044-A

**5. MONITORING ORGANIZATION REPORT NUMBER(S):** ARO 24567.2-EL

**6a. NAME OF PERFORMING ORGANIZATION:** J.S. Lee Associates, Inc.

**6b. OFFICE SYMBOL** *(If applicable)*:

**7a. NAME OF MONITORING ORGANIZATION:** U. S. Army Research Office

**6c. ADDRESS** *(City, State, and ZIP Code)*:
2001 Jefferson Davis Hwy., Suite 601
Arlington, VA 22202

**7b. ADDRESS** *(City, State, and ZIP Code)*:
P. O. Box 12211
Research Triangle Park, NC 27709-2211

**8a. NAME OF FUNDING / SPONSORING ORGANIZATION:** U. S. Army Research Office

**8b. OFFICE SYMBOL** *(If applicable)*:

**9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:** DAAL03-87-C-0006

**8c. ADDRESS** *(City, State, and ZIP Code)*:
P. O. Box 12211
Research Triangle Park, NC 27709-2211

**10. SOURCE OF FUNDING NUMBERS**

| PROGRAM ELEMENT NO. | PROJECT NO. | TASK NO. | WORK UNIT ACCESSION NO. |
|---|---|---|---|
| | | | |

**11. TITLE** *(Include Security Classification)*:
The Diversity ECCM Performance of Frequency-Hopping CPFSK in Partial-Band Noise Jamming

**12. PERSONAL AUTHOR(S):** L. E. Miller, R. H. French, J. S. Lee, H. M. Kwon

**13a. TYPE OF REPORT:** Final

**13b. TIME COVERED:** FROM 5/1/87 TO 4/30/88

**14. DATE OF REPORT** *(Year, Month, Day)*: 88 May 25

**15. PAGE COUNT:** xii+197

**16. SUPPLEMENTARY NOTATION:** The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

**17. COSATI CODES**

| FIELD | GROUP | SUB-GROUP |
|---|---|---|
| | | |
| | | |

**18. SUBJECT TERMS** *(Continue on reverse if necessary and identify by block number)*:
Frequency-hopping, digital FM radio, ECCM, jamming, statistical communication theory, phase distributions

**19. ABSTRACT** *(Continue on reverse if necessary and identify by block number)*

A thorough analysis is made of the uncoded probability of error achieved by a hopped CPFSK (narrowband digital FM) radio link while subject to partial-band noise jamming. Included in the analysis are the effects of thermal noise, FM clicks, and intersymbol interference, in addition to the jamming. For both limiter-discriminator and differential detection, it is shown that a repeat diversity combining method of summing detector samples does not yield a diversity gain against worst-case partial-band noise jamming, despite the nonlinearity of the receiver. The analytical procedures formulated, and computer programs developed, can be modified to treat other CPFSK combining metrics; it is recommended that other soft-decision metrics be evaluated for ECCM properties, since it is known that a diversity gain is theoretically possible using perfect side information. In the report, easily calculated approximations and short-cut expressions for the error probability are assessed for accuracy, using the more complicated exact formulas. Computer programs are included in appendices.

**20. DISTRIBUTION / AVAILABILITY OF ABSTRACT:** ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS

**21. ABSTRACT SECURITY CLASSIFICATION:** Unclassified

**22a. NAME OF RESPONSIBLE INDIVIDUAL:** Dr. William A. Sander

**22b. TELEPHONE** *(Include Area Code)*: (919) 549-0641

**22c. OFFICE SYMBOL:** ARO: EL-S

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

JC-2044-A

# THE DIVERSITY ECCM PERFORMANCE OF FREQUENCY-HOPPING CPFSK IN PARTIAL-BAND NOISE JAMMING

FINAL REPORT

MAY 25, 1988

PREPARED FOR

U.S. ARMY RESEARCH OFFICE

CONTRACT

DAAL03-87-C-0006

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

J. S. LEE ASSOCIATES, INC.
2001 JEFFERSON DAVIS HIGHWAY, SUITE 601
ARLINGTON, VIRGINIA 22202

Approved for public release; distribution unlimited.

TABLE OF CONTENTS



DTIC
COPY
INSPECTED
6

on For
RA&I
ced
ation
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Concluded)

LIST OF FIGURES

LIST OF FIGURES (Cont.)

LIST OF FIGURES (Concluded)

## LIST OF TABLES

# THE DIVERSITY ECCM PERFORMANCE OF FREQUENCY-HOPPING CPFSK IN PARTIAL-BAND NOISE JAMMING

## 1.0 INTRODUCTION

### 1.1 BACKGROUND

While frequency-hopped MFSK (M-ary frequency-shift keying) waveforms are widely discussed as candidates for spread spectrum applications, there are many current applications in which the chosen hop modulation is narrowband digital FM, or CPFSK (continuous phase FSK) in its many forms, most notably Army tactical radios. In addition to offering efficient use of the available spectrum, it has been estimated that limiter-discriminator detection of a hopped CPFSK waveform can obtain a 4 dB improvement in performance over MFSK systems in noise jamming [21]. However, the few published analyses giving results for FH/CPFSK are either based on approximations or neglect thermal noise. We know of none that address tone jamming of FH/CPFSK.

For the optimal utilization and design of these systems, it is desirable to discover what parameter values work best under jamming. Figure 1.1-1 illustrates a slow-hopping FH/CPFSK system without diversity. Shown in the figure are optional transmitter elements associated with transmission bandwidth control. The primary transmission parameter is the normalized maximum frequency deviation, $h = 2f_dT$, where T is the bit time. The signal bandwidth is proportional to h. At the receiver, an I.F. (intermediate frequency) filter excludes unwanted receiver conversion products and controls the amount of noise admitted. If the bandwidth, $W_{IF}$, of this filter is large, the received FM waveform passes through to the detector undistorted. If the bandwidth is decreased, less noise is admitted but the resulting distortion of the waveform gives

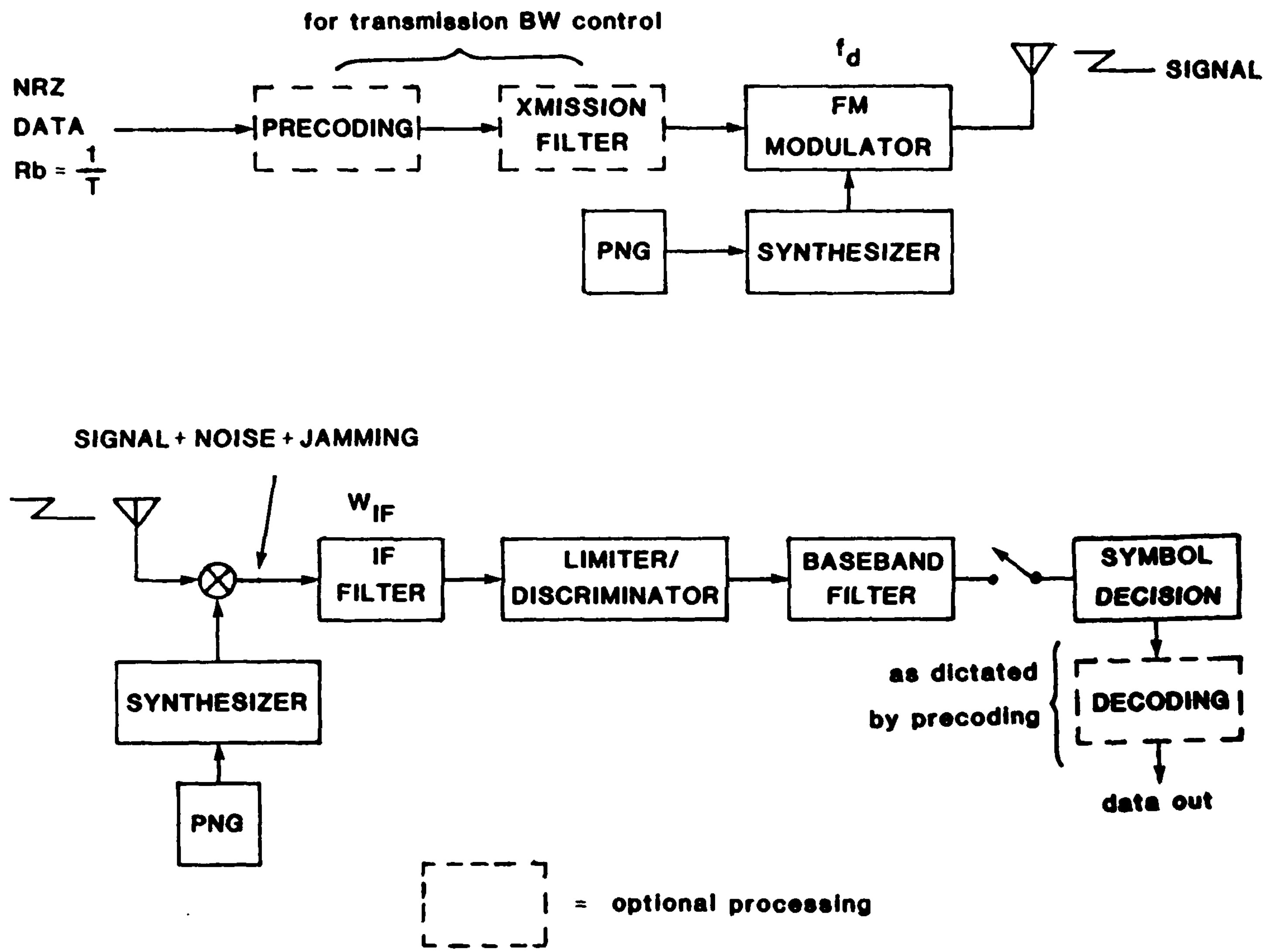



FIGURE 1.1-1. FH/CPFSK SYSTEM

rise to intersymbol interference. Thus there are optimum values of h and the product $W_{IF}T$ which are traded off against one another. Without hopping or jamming, it has been shown the h = 0.7 and $W_{IF}T$ = 1.0 are best values [3]. With worst-case partial-band jamming but neglecting thermal noise, it has been asserted that h = 0.6 and $W_{IF}T$ = 0.75 are best values [22].

## 1.2 RATIONALE FOR THE STUDY

We have mentioned previously that detailed analyses of hopped CPFSK systems under various kinds of jamming and also system or thermal noise are not available in the current literature. Because of the potential advantages offered by CPFSK over MFSK and because many current hopped and unhopped radio systems employ some form of CPFSK, it is important to expand the knowledge of the performance that can be expected from FH/CPFSK systems, and what parameter values are optimum.

Noncoherent detection of CPFSK can be performed using a limiter-discriminator with integrate-and-dump filtering, or by differential detection. In either case, a difference $\Delta\Phi$ in the total I.F. waveform phase is extracted which, in the absence of noise and/or distortion, would convey the transmitted binary information. Normally, that is, without repetition or diversity, a hard decision is made based on the sign of $\Delta\Phi$. Typically $\Delta\Phi$ is corrupted by noise projected onto the phase of the signal by the action of the bandpass limiter which precedes the discriminator. Now, since the phase of a carrier is inherently ambiguous (modulo $2\pi$), it can be written

$$\Delta\Phi = (\Delta\Phi)\ \mathrm{mod}2\pi + 2\pi N, \qquad (1.2\text{-}1)$$

where N is the net number of positive $2\pi$ phase rotations or "clicks" in the bit interval; usually N is taken to be a negative random (Poisson) integer if the data gives $\Delta\Phi > 0$, and positive if $\Delta\Phi < 0$. For high carrier-to-noise ratios (CNR), the probability that N is nonzero becomes very small. It has been argued then that the limiter-discriminator detection, in effect, limits or clips the phase noise, and therefore that jamming effects in an FH/CPFSK system will be suppressed. Our study tests this hypothesis by postulating and evaluating an FH/CPFSK system with soft-decision diversity combining of received $\Delta\Phi$ samples, implementing the decision rule

$$\mathrm{SUM}_k \left\{(\Delta\Phi)_k\right\} \underset{\text{bit} = 0}{\overset{\text{bit} = 1}{\gtrless}} 0. \tag{1.2-2}$$

In our analysis and computations forming the body of this report, we follow a rigorous treatment of the system, including background noise, intersymbol interference, and FM noise clicks. We consider linear combining of receiver samples for both limiter-discriminator and differential detection types of CPFSK receivers, since both are used in current tactical FH/CPFSK radios.

## 1.3 SYSTEM STUDIED

In view of the importance of the FH/CPFSK waveform to tactical military communications, and of the potential for diversity improvements in hopping system performance against jamming, we have undertaken the studies summarized in this report. The following material briefly describes the system studied.

### 1.3.1 Transmission Scheme

Our studies concern the jammed performance of hopped binary FM communications, particularly under the assumption of partial-band noise jamming and the use of (time) diversity to mitigate the jamming. Figure 1.3-1 gives a block diagram of the transmission scheme for the system. Binary data, after error-control coding, is to be transmitted using slow-frequency-hopping digital FM, or CPFSK. The coded symbols are to be repeated on L different hops in order to increase the likelihood that some of the symbols are free of jamming. The figure suggests one of many possible ways to accomplish this objective. According to the version shown in the figure, the coded symbols are first read into a Q-bit shift register (Q-symbol buffer), where Q is the number of symbols that can be transmitted in one hop period. For example, if the channels allotted to the system support 20 kbps digital FM signalling, and the hop rate is 100 hops/sec, then Q could be 200.

When the Q symbols have all been generated at rate $R_c$ and stored in the input buffer, they are then transferred to a second (output) buffer. The transmitter logic reads this buffer L times at the rate $LR_c$, and this stream

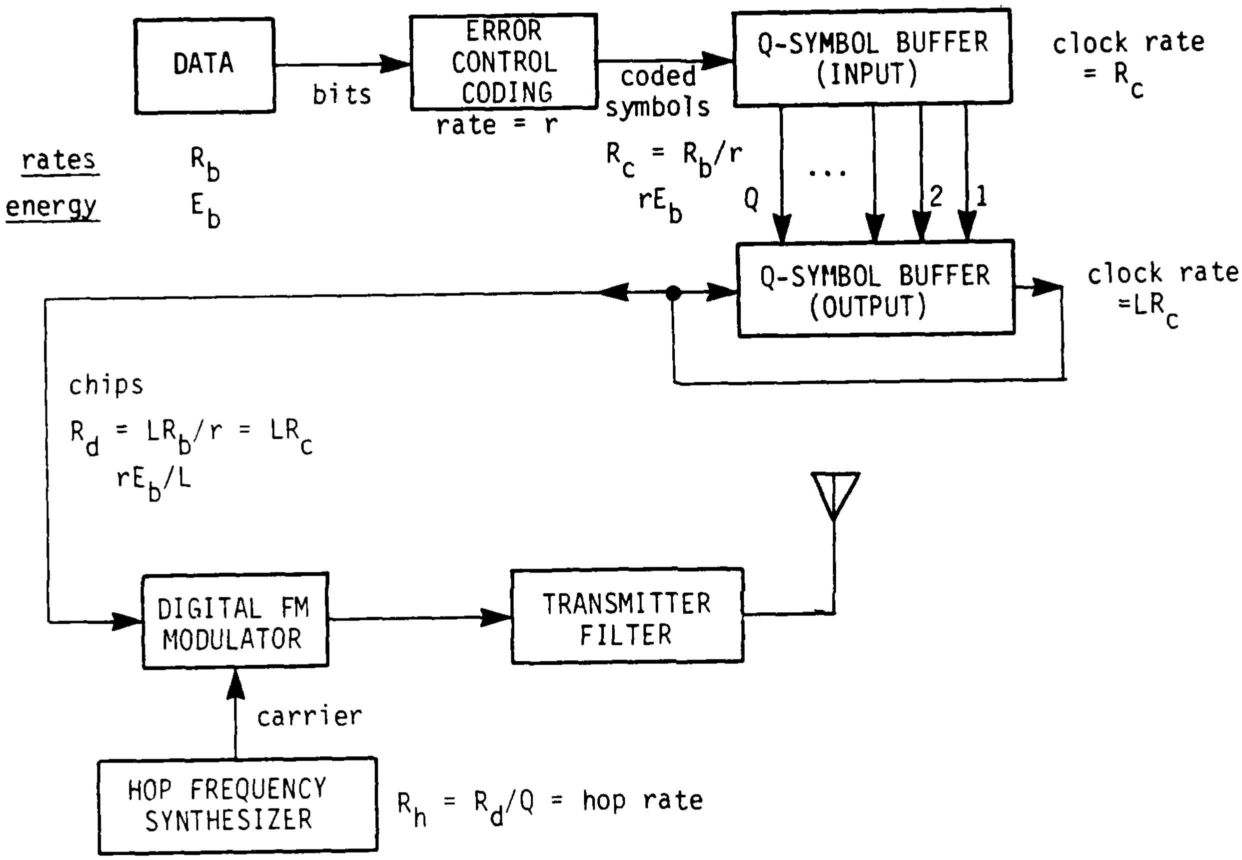



FIGURE 1.3-1 TRANSMISSION SCHEME

of data "chips" is used to frequency-modulate the selected carrier frequency, which is changed (hopped) to a new, pseudorandomly-selected value after Q chips have been transmitted. In this manner, L copies of the Q-symbol sequence have been transmitted on L different, successive hops. Although Figure 1.3-1 suggests that the Q symbols are repeated in the same order on each hop, it is of course possible with a more sophisticated system to permute the symbols or otherwise scramble them so that the ordering of the symbols is different on each hop.

We note that certain fundamental relationships exist among the digital rates at various points in the transmission logic, and among the energies in the transmitted waveform which correspond to each rate. The symbols actually transmitted are keyed at the rate $R_d$ which, as we have already noted, is a basic specification of the communications channels being used for hopping. The original symbol rate is $R_c = R_d/L$, on account of the repetitions. Viewed another way, the energy transmitted per chip is the fraction 1/L of the coded symbol energy. If the error control code rate is r, then the original bit rate is

$$R_b = R_c r = r\, R_d/L. \tag{1.3-1}$$

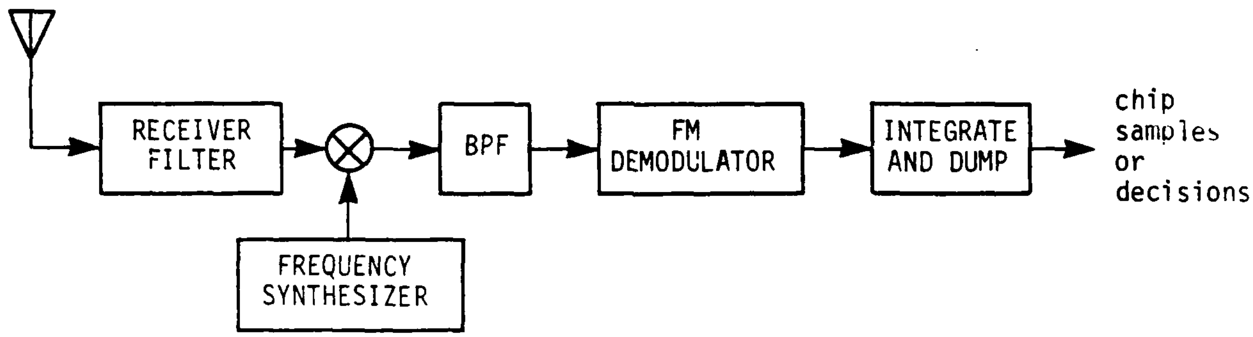
For example, for $R_d$ = 20 kbps, r = 1/2, and L=5, $R_c$ = 4 kbps; between the coding and the repetitions, the bit energy gets split up into 10 pieces in this example.

### 1.3.2 Reception scheme

It is obvious that synchronization and timing are especially important to the type of waveform we are considering. The receiver, diagrammed in

Figure 1.3-2, must accurately synthesize local oscillator frequencies in order to dehop the signal at the proper times, and must then be capable of sampling the demodulator output at the ends of each chip interval. These samples, QL of them for L complete hops, are buffered so that the receiver logic can in some manner combine the L chips belonging to a particular code symbol.

As the figure suggests, binary decisions can be made on each chip as it is received. The resulting logic and buffering for such a "hard decision" procedure is simpler than that required for a "soft decision" scheme, which involves A/D conversion of the samples and storing the resulting QL multi-bit words, one per chip. With either hard or soft chip processing, after diversity combining there is the option to do binary decoding of code symbol decisions, or soft-decision decoding of code symbol metrics produced by the combining.




(a) Chip Sampling


(b) Diversity Combining and Decoding

FIGURE 1.3-2 RECEPTION SCHEME

## 1.4 SUMMARY AND RECOMMENDATIONS

### 1.4.1 Main Conclusion

Our work thus far on FH/CPFSK systems with diversity, documented in this report, has determined that linear combining or summing of the diversity components (chips) of each bit does not yield an improvement in uncoded system performance against worst-case partial-band noise jamming. This conclusion holds for both limiter-discriminator and differential detection techniques.

### 1.4.2 Achievements

In developing analytical and computational methods for the calculations which produced this conclusion, we have made considerable advances in the methodology for evaluating FH/CPFSK and CPFSK systems. These include:

(a) Independent derivations of both integral and series forms of the differential phase probability density function, showing explicitly the effect of taking the phase to be modulo $2\pi$.

(b) Explicit derivations and formulas for intersymbol interference-related SNR and differential phase parameters for all eight of the possible adjacent bit data patterns, and arbitrary I. F. filter transfer functions.

(c) An independent and detailed derivation of FM noise click rates and average number of positive clicks, and development of the notion of "significant clicks" in computation of the error probability.

(d) A general formulation of the L hop/bit FH/CPFSK jammed error probability valid for any diversity combining technique.

(e) An original derivation of the diversity sum jammed error probability for FH/CPFSK using differential detection.

(f) Comparison of exact and simplified performance calculations (the exact require much computer time and effort; the simplified can be done on a programmable calculator).

(g) Development of a computer program based on the discrete Fourier transform for evaluating the probability of error, which is useful for all sum metrics.

(h) Demonstration that, if perfect side information is used, there is a diversity gain for FH/CPFSK.

### 1.4.3 Recommendations

The diversity sum method for combining the L chips for a given FH/CPFSK bit can be regarded as implementing a form of soft-decision metric. Note that the summing operation does not utilize any side information on whether a particular chip is jammed, or how strongly it is jammed. Since there is no mechanism for excluding or otherwise treating jammed chips differently, when one or more chips are subject to jamming, the entire metric is corrupted, and this accounts for the failure of the linear combining (simple sum) metric to provide a diversity improvement.

Now, it is a fact that if instead of soft-decision combining of the chips, we combine hard decisions on each chip, for high $E_b/N_0$ (15 dB or more) a diversity gain results, in the sense that for a particular J/S ratio $L > 1$ may yield a lower bit error probability. The improvement is due to the hard decisions' limiting a jammed chip to "one vote" in the sum of hard decisions. Since no side information is required, this simple ECCM scheme is very attractive, except for the fact that noncoherent combining losses are high for hard-decision metrics, relative to soft decisions.

It can be shown that a "perfect side information" soft-decision scheme which includes only unjammed chips in the sum can provide greater diversity gain than the hard-decision combining strategy. Therefore, some practical method which combines analog chip samples or soft decisions is likely to exist for improving FH/CPFSK performance in partial-band jamming more than the hard-decision metric does. Such metrics have been found for noncoherent FH/BFSK [23,24], including a "self-normalizing" technique which does not require side information. Therefore, we recommend that further studies of the type given in this report be conducted of soft-decision diversity combining metrics for FH/CPFSK which have the potential for improved performance in jamming.

## 2.0 MODELLING AND PRELIMINARY ANALYSES

In this section, we describe the modelling assumptions used to analyze the performance of frequency-hopped continuous phase frequency-shift keying (FH/CPFSK) in the presence of thermal noise and partial-band noise jamming. Limiter-discriminator (LD) detection is assumed. Certain preliminary analyses are conducted to predict the effects of intersymbol interference (ISI) on the detected waveform, and to derive the distribution of the noise projected onto the signal phase.

### 2.1 DESCRIPTION OF SYSTEM WAVEFORMS

For discussion purposes, we consider first the simplified system model depicted in Figure 2.1-1.

#### 2.1.1 Overview of System Operation

A binary data source outputting the sequence $\{d_k\}$ modulates an FM transmitter with the NRZ bipolar data waveform d(t), where

$$d(t) = \sum_k d_k\, p(t-kT),\ d_k = \pm 1, \tag{2.1-1}$$

and the pulse function p(t) is assumed to be rectangular:

$$\begin{aligned} p(t) &= u(t)-u(t-T) \\ &= \mathrm{rect}(t-T/2), \end{aligned} \tag{2.1-2}$$

with u(t) the unit step function. The interval T is the bit duration, so that the data rate is $R_b = 1/T$. Although much attention is being directed

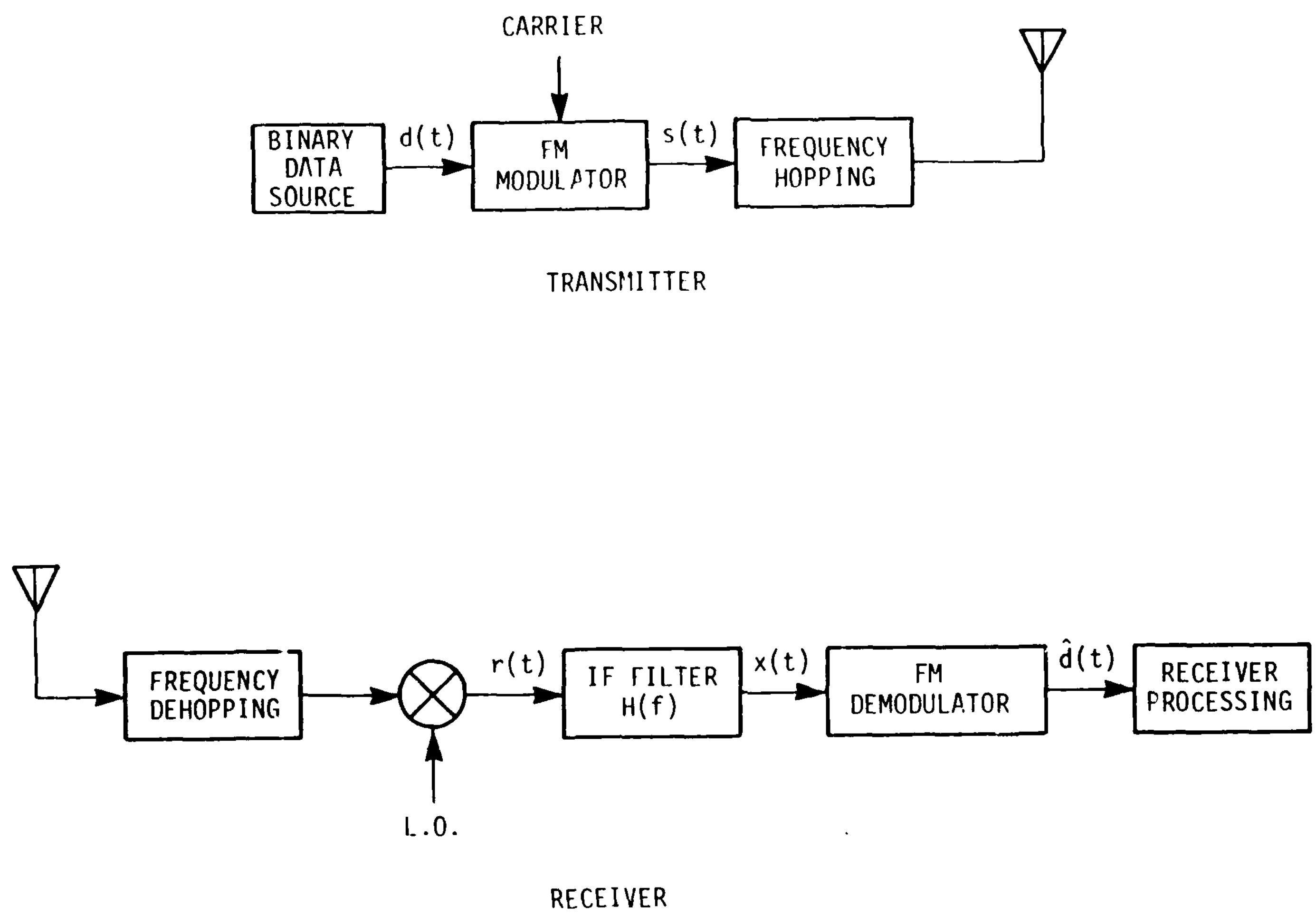



FIGURE 2.1-1 SIMPLIFIED MODEL OF AN FH/CPFSK SYSTEM

currently at reducing the transmitted bandwidth of binary FM signals by employing non-rectangular p(t) and by correlative coding of the data, here we will assume the p(t) as given and will treat the $d_k$ bit values as having been generated independently. (A more detailed discussion of the generation of the data will be presented below.)

The resulting modulated carrier frequency is given by

$$f(t) = f_0 + f_d \cdot d(t), \tag{2.1-3}$$

and the commonly accepted measure of relative frequency deviation is the modualtion index

$$h \doteq 2f_d/R_b = 2f_d T. \tag{2.1-4}$$

Thus the emitted waveform (neglecting any *post-modulator filters or distortion*, and without frequency hopping of the carrier $f_0$) is

$$s(t) = \text{const} \cdot \cos\left\{ 2\pi f_0 t + 2\pi f_d \int_{-\infty}^{t} d\tau \; d(\tau) \right\}. \tag{2.1-5}$$

Ideally, the frequency hopping and dehopping operations shown in Figure 2.1-1 are in perfect synchronism (with the propagation delay accounted for), and are therefore "transparent" in the sense that whether the carrier is fixed or hopped should not affect system operation.

Against a noise jammer occupying a portion of the RF band over which the system is hopping, the hopping and dehopping result in the following total waveform prior to intermediate frequency (I.F.) filtering, overlooking harmonics to be rejected by that filtering:

$$r(t) = A\cos\{2\pi f_0 t + \theta_0 + \theta_m(t)\} + n(t). \qquad (2.1\text{-}6)$$

In this formulation, we show an undistorted signal term; $\theta_0$ is a random phase offset left over from the downconversion and dehopping, and for convenience we define

$$\theta_m(t) \triangleq 2\pi f_d \int_{-\infty}^{t} d\tau\, d(\tau). \qquad (2.1\text{-}7)$$

The noise term $n(t)$ is given by

$$n(t) = \begin{cases} n_0(t), & \text{hop was not jammed} \\ n_0(t) + n_J(t), & \text{hop was jammed.} \end{cases} \qquad (2.1\text{-}8)$$

The background and receiver additive bandpass noise term $n_0(t)$ and the jamming bandpass noise term $n_J(t)$ are assumed to have flat spectra going into the I.F. filter, with two-sided spectrum levels $N_0/2$ and $N_{0J}/2$, respectively.

Conceptually, the received signal - with more or less noise, depending on whether the signal was jammed or not on a particular hop - is filtered and demodulated to recover an estimate $\hat{d}(t)$ of the data waveform $d(t)$ sent originally by the transmitter. The information in $\hat{d}(t)$ is then extracted by receiver processing.

Now we must go into considerably more detail to describe the structure of the data sequence and its relation to the hopping scheme. Following that, we discuss the distortion effects of receiver filtering and describe the receiver processing in further detail.

2.1.2 The I.F. Filter Output, Signal Term

The I.F. filtering operation is intended to reject all but the selected signal. Its finite bandwidth is taken to be its noise bandwidth $W_{IF} = 2B_0$, where the bandpass filter transfer function $H(f)$ is modelled in terms of a lowpass filter transfer function $H_0(f)$ by the relation

$$H(f) = H_0(f-f_0) + H_0(f+f_0); \tag{2.1-9a}$$

that is, its impulse response is considered to be

$$h(t) = 2h_0(t) \cos \omega_0 t, \tag{2.1-9b}$$

and $h_0(t)$ is the lowpass filter impulse response corresponding to $H_0(f)$. Accordingly, the noise bandwidth $B_0$ is given by

$$B_0 = \int_0^\infty df\,|H_0(f)|^2 \Big/ |H_0(0)|^2 \ . \tag{2.1-10}$$

For example, if the lowpass filter has the Gaussian shape

$$H_0(f) = e^{-\pi f^2/8B_0^2} , \tag{2.1-11}$$

then the 3 dB bandwidth $B_3$ is 0.9294 times the noise bandwidth $B_0$. For an n-pole Butterworth filter with transfer function

$$|H_0(f)|^2 = [1 + (f/B_3)^{2n}]^{-1}, \tag{2.1-12a}$$

the noise bandwidth is related to the 3 dB bandwidth by

$$\begin{aligned} B_0 &= B_3 \, (\pi/2n) \Big/ \sin(\pi/2n) \\ &= 1.5708\, B_3, \; n = 1 \\ &= 1.1107\, B_3, \; n = 2 \\ &= 1.0472\, B_3, \; n = 3 \\ &= 1.0262\, B_3, \; n = 4. \end{aligned} \tag{2.1-12b}$$

Now, since, theoretically, the FM signal waveform has infinite bandwidth the I.F. filtering rejects not only unwanted signals but also portions of the desired signal. Therefore the filtering introduces distortion; for the common *modelling assumptions* we are making, in effect the distortion caused by the finite I.F. bandwidth represents all the distortion suffered by the waveform - at least all distortion due to filtering - just as the noise at I.F. represents all noise present.

The signal part of the filter output is found explicitly by the following development for the signal term (using (*) to denote convolution):

$$\begin{aligned} s(t)*h(t) &= \int_{-\infty}^{t} d\tau \; 2h_0(t-\tau) \cos\omega_0(t-\tau) \cdot A \cos[\omega_0\tau + \theta_0 + \theta_m(\tau)] \\ &= 2A \int_{-\infty}^{t} d\tau \; h_0(t-\tau) \cos\theta_m(\tau) \cos\omega_0(t-\tau) \cos(\omega_0\tau+\theta_0) \\ &\quad -2A \int_{-\infty}^{t} d\tau \; h_0(t-\tau) \sin\theta_m(\tau) \cos\omega_0(t-\tau) \sin(\omega_0\tau+\theta_0) \end{aligned}$$

$$= A\int_{-\infty}^{t} d\tau\ h_0(t-\tau)\ \cos\theta_m(\tau)\ \{\cos(\omega_0 t+\theta_0)+\cos[\omega_0(2\tau-t)+\theta_0]\}$$

$$-A\int_{-\infty}^{t} d\tau\ h_0(t-\tau)\ \sin\theta_m(\tau)\{\sin(\omega_0 t+\theta_0)+\sin[\omega_0(2\tau-t)+\theta_0]\}$$

$$\approx A\cos(\omega_0 t+\theta_0)\int_{-\infty}^{t} d\tau\ h_0(t-\tau)\ \cos\theta_m(\tau)$$

$$-\ A\sin(\omega_0 t+\theta_0)\int_{-\infty}^{t} d\tau\ h_0(t-\tau)\ \sin\theta_m(\tau)$$

$$= a(t)\ A\cos[\omega_0\ t+\theta_0+\phi(t)],\quad f_0 >> B, \tag{2.1-13}$$

where

$$a^2(t) = [h_0(t) * \cos\theta_m(t)]^2 + [h_0(t) * \sin\theta_m(t)]^2 \tag{2.1-14a}$$

and

$$\phi(t) = \tan^{-1}\{[h_0(t) * \sin\theta_m(t)]/[h_0(t) * \cos\theta_m(t)]\}\ . \tag{2.1-14b}$$

From this development we see that the filter distorts the signal phase waveform from $\theta_m(t)$ to $\phi(t)$, and induces amplitude modulation $a(t)$.

For most cases of practical interest, it is sufficient to consider intersymbol interference effects (i.e., the overlapping of filter responses from different bit intervals) due to immediately adjacent bits [1]. Therefore, in what follows we consider bit patterns which are the periodic extensions of the patterns

$1\underline{1}1$, $0\underline{0}0$ (all one's or zeros); (2.1-15a)

$0\underline{1}0$, $1\underline{0}1$ (alternating one's and zeros); (2.1-15b)

and $0\underline{1}10$, $1\underline{0}01$, $1\underline{1}00$, $0\underline{0}11$. (2.1-15c)

The "present bit" in these sequences is indicated by the underlining. The patterns in (2.1-15) were chosen because they generate the eight possible 3-bit patterns in a very simple manner and can be analyzed easily.

Using the steady-state filtering approach of Tjhung and Wittke [2] and Pawula [3], we recognize that if the Fourier series for the periodic extension of the i:th patterns yields (assuming evenness of $\theta_m$ about $t = 0$)

$$\sin\theta_m^{(i)}(t) = \sum_{k=1}^{\infty} \alpha_k^{(i)} \cos(2\pi k f_p t) \tag{2.1-16a}$$

and

$$\cos\theta_m^{(i)}(t) = \sum_{k=1}^{\infty} \beta_k^{(i)} \cos(2\pi k f_p t) \tag{2.1-16b}$$

where $f_p$ is the appropriate fundamental frequency of $\theta_m^{(i)}(t)$, then the responses of the lowpass filter $h_0(t)$ to these components are

$$u_i(t) \triangleq h_0(t)*\sin\theta_m^{(i)}(t)$$

$$= \sum_{k=1}^{\infty} |H_0(kf_p)|\alpha_k^{(i)} \cos[2\pi k f_p t - B(kf_p)] \tag{2.1-17a}$$

and $v_i(t) \triangleq h_0(t) * \cos\theta_m^{(i)}(t)$

$$= \sum_{k=1}^{\infty} |H_0(kf_p)|\beta_k^{(i)} \cos[2\pi k f_p t - B(kf_p)] \tag{2.1-17b}$$

where $B(f)$ is the filter phase delay. For convenience, we shall employ the alternative notations

$$u(t;\text{pattern } i) \equiv u_i(t) \tag{2.1-17c}$$

$$v(t;\text{pattern } i) \equiv v_i(t)\ . \tag{2.1-17d}$$

Note from (2.1-15) that

$$\tan\phi(t) = u(t)/v(t) \tag{2.1-18}$$

for each pattern.

For patterns $1\underline{1}1$ and $0\underline{0}0$, the signal is a pure sinusoid, so that (neglecting any filter delay) the quadrature components of the signal are

$$u(t;1\underline{1}1) = -u(t;0\underline{0}0) = a_0 \sin(\pi ht/T) \tag{2.1-19a}$$

and

$$v(t;1\underline{1}1) = v(t;0\underline{0}0) = a_0 \cos(\pi ht/T). \tag{2.1-19b}$$

That is, for these two patterns, $a(t) = a_0$ and $\phi(t) = \pm ht/T$, where

$$a_0 \triangleq |H_0(f_d)| \ . \tag{2.1-19c}$$

For patterns $0\underline{1}0$ and $1\underline{0}1$, the original frequency modulation is a $\pm f_d$ squarewave with period 2T, so that $\theta_m(t)$ is a bipolar triangular wave with amplitude $\pi h/2$ and period 2T. For pattern $0\underline{1}0$, the triangular wave's positive peak occurs at t = 0, using the convention that $d(t) = d_k$ for $(k-1)T < t < kT$. Expanding $\sin \theta_m(t)$ and $\cos \theta_m(t)$ in Fourier series gives

$$\begin{aligned} \sin\theta_m(t;0\underline{1}0) &= -\sin\theta_m(t;1\underline{0}1) \\ &= \frac{4h}{\pi} \cos\left(\frac{\pi h}{2}\right) \sum_{k=1}^{\infty} \frac{\cos[(2k-1)\pi t/T]}{(2k-1)^2-h^2} \end{aligned} \tag{2.1-20a}$$

and

$$\begin{aligned} \cos\theta_m(t;0\underline{1}0) &= \cos\theta_m(t;1\underline{0}1) \\ &= \frac{2}{\pi h} \sin\left(\frac{\pi h}{2}\right)\left[1 - 2h^2 \sum_{k=1}^{\infty} \frac{\cos(2k\pi t/T)}{(2k)^2-h^2}\right]. \end{aligned} \tag{2.1-20b}$$

Assuming that the I.F. filter passes only harmonics of these Fourier series up to f = 1/T, we find that

$$u(t;0\underline{1}0) = -u(t;1\underline{0}1)$$

$$\simeq \frac{4h}{\pi}\cos(\frac{\pi h}{2})|H_0(\frac{1}{2T})|\ \frac{1}{1-h^2}\cos[\frac{\pi t}{T} - B(\frac{1}{2T})] \qquad (2.1\text{-}21a)$$

and

$$v(t;0\underline{1}0) = v(t;1\underline{0}1)$$

$$\simeq \frac{2}{\pi h}\sin(\frac{\pi h}{2})\left\{1 - \frac{2h^2}{4-h^2}|H_0(\frac{1}{T})|\cos[\frac{2\pi t}{T} - B(\frac{1}{T})]\right\}, \qquad (2.1\text{-}21b)$$

where the phase delay B(f) is taken to be zero if a Gaussian-shaped filter is assumed. Figure 2.1-2 illustrates the various waveforms associated with this type of pattern.

For bit patterns $0\underline{1}10$ and $1\underline{0}01$, the original frequency modulation is a $\pm f_d$ squarewave with period 4T, so that $\theta_m(t)$ is a bipolar triangular wave with amplitude $\pi h$ and period 4T. For pattern $0\underline{1}10$, the triangular wave's positive peak occurs at t = T. By analogy with (2.1-20), the Fourier series expansions for $\sin\theta_m(t)$ and $\cos\theta_m(t)$ are

$$\sin\theta_m(t;1\underline{1}00) = -\sin\theta_m(t;0\underline{0}11)$$

$$= \frac{8h}{\pi}\cos(\pi h)\sum_{k=1}^{\infty}\frac{\cos[(2k-1)\pi(t-T)/2T]}{(2k-1)^2-4h^2}, \qquad (2.1\text{-}22a)$$

and $\cos\theta_m(t;1\underline{1}00) = \cos\theta_m(t;0\underline{0}11)$

$$= \frac{\sin(\pi h)}{\pi h}\left[1 - 2h^2\sum_{k=1}^{\infty}\frac{\cos[k\pi(t-T)/T]}{k^2-h^2}\right]. \qquad (2.1\text{-}22b)$$

Assuming the filter rejects harmonics with f > 1/T, we find that the signal quadrature components for these patterns are

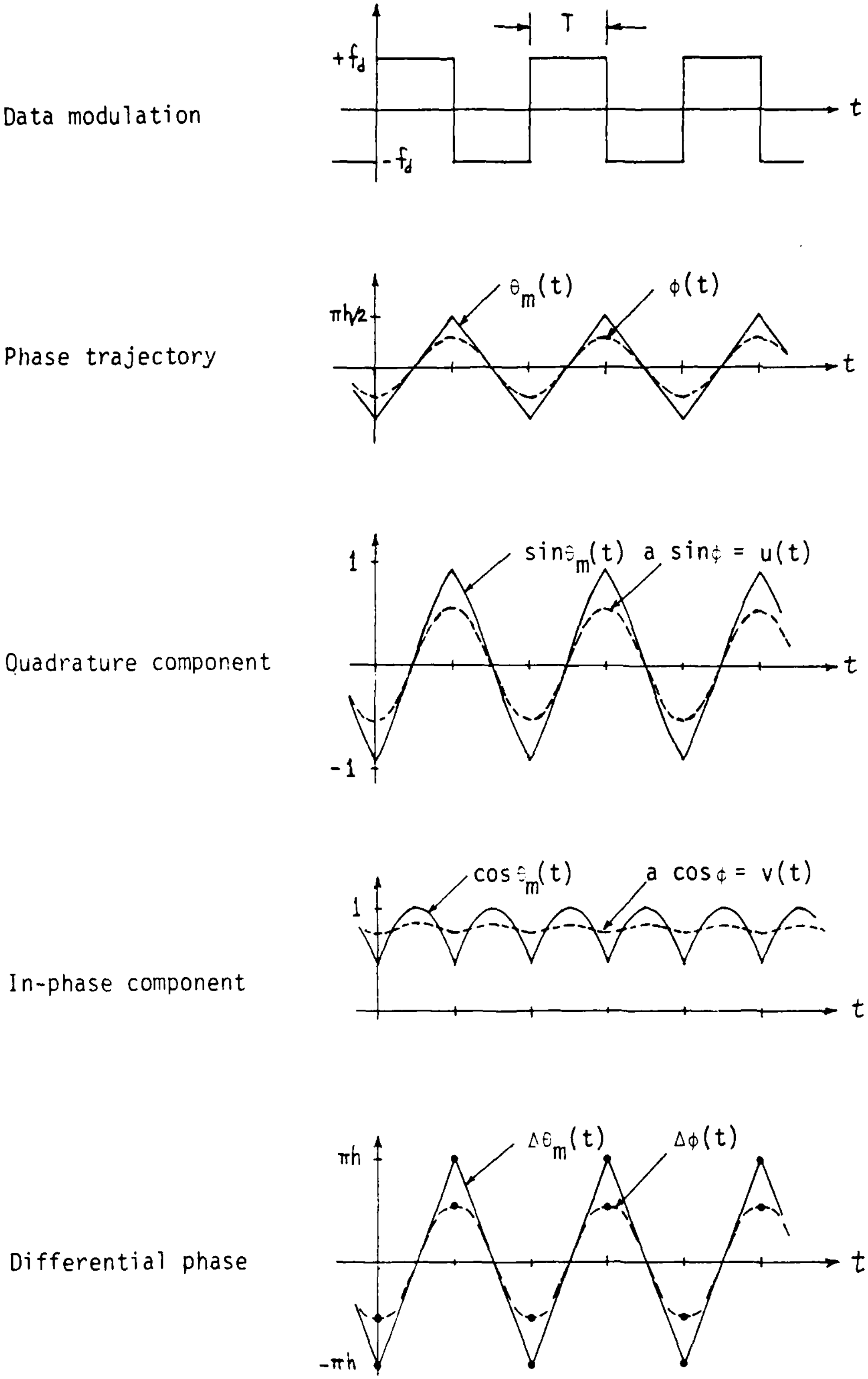



FIGURE 2.1-2. WAVEFORMS ASSOCIATED WITH ALTERNATING 1's AND 0's PATTERNS.

$$u(t;1\underline{1}00) = -u(t;0\underline{0}11)$$

$$\simeq \frac{8h}{\pi}\cos(\pi h)\left\{\frac{1}{1-4h^2}\left|H_0\left(\frac{1}{4T}\right)\right|\sin[\pi t/2T-B(1/4T)]\right.$$

$$\left.-\frac{1}{9-4h^2}\left|H_0\left(\frac{3}{4T}\right)\right|\sin[3\pi t/2T-B(3/4T)]\right\} \qquad (2.1\text{-}23a)$$

and

$$v(t;1\underline{1}00) = v(t;0\underline{0}11)$$

$$\simeq \frac{\sin(\pi h)}{\pi h}\left\{1+\frac{2h^2}{1-h^2}\left|H_0\left(\frac{1}{2T}\right)\right|\cos[\pi t/T-B(1/2T)]\right.$$

$$\left.-\frac{2h^2}{4-h^2}\left|H_0\left(\frac{1}{T}\right)\right|\cos[2\pi t/T-B(1/T)]\right\}. \qquad (2.1\text{-}23b)$$

Figure 2.1-3 illustrates the various waveforms associated with this type of pattern.

Recognizing that the time extensions of the patterns $0\underline{1}10$ and $1\underline{0}01$ are time shifted versions of those for $1\underline{1}00$ and $0\underline{0}11$, we can immediately write

$$u(t;0\underline{1}10) = -u(t;1\underline{0}01)$$

$$= u(t+T;1\underline{1}00) \qquad (2.1\text{-}24a)$$

and

$$v(t;0\underline{1}10) = v(t;1\underline{0}01)$$

$$= v(t+T;1\underline{1}00). \qquad (2.1\text{-}24b)$$

A summary of these components which determine $\phi(t)$ is given in Table 2.1-1, assuming no filter delay, with example values as listed in Table 2.1-2 for $h = 0.7$ and $D \triangleq W_{IF}T = 1.0$.

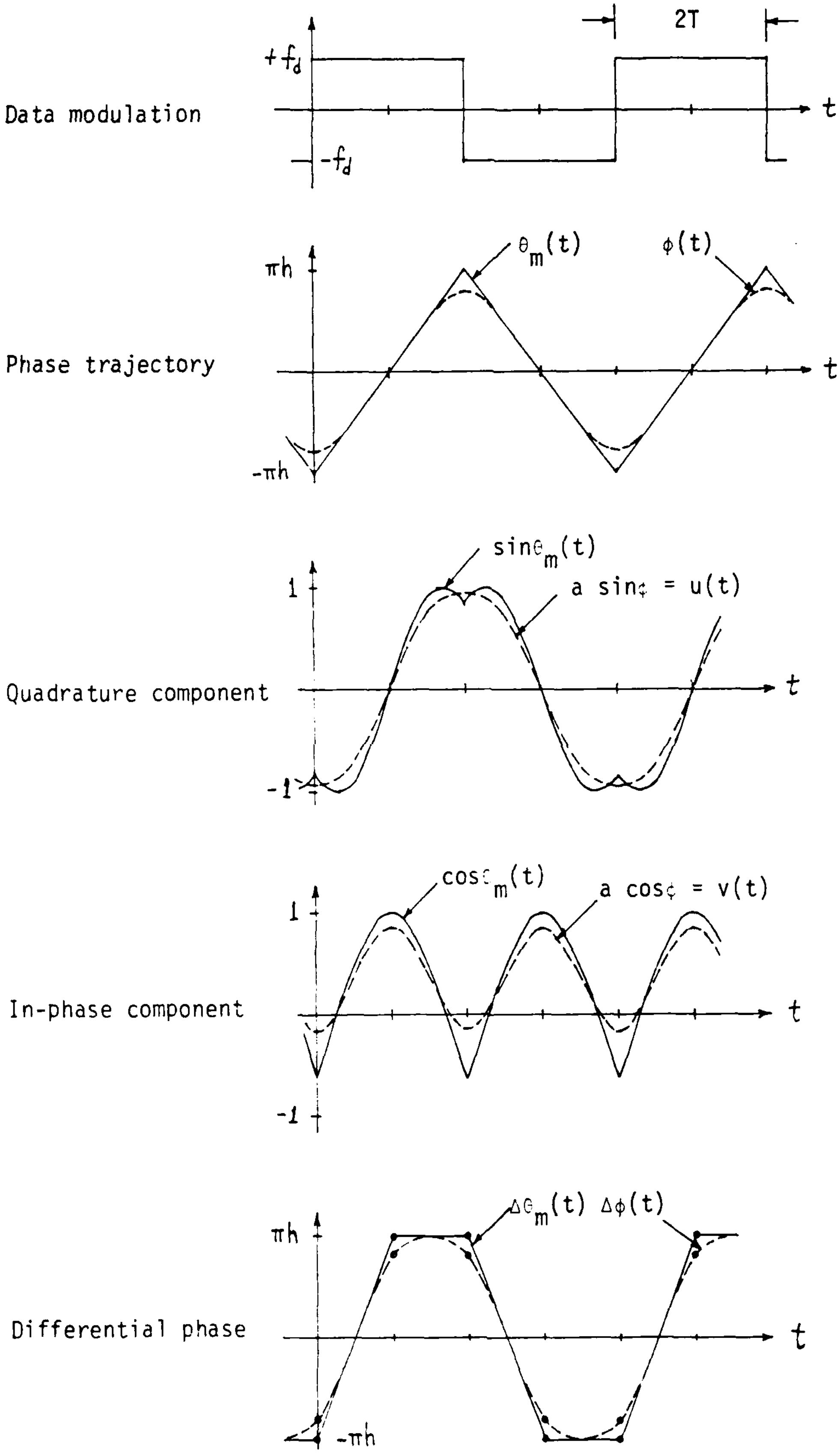



FIGURE 2.1-3. WAVEFORMS ASSOCIATED WITH ALTERNATING PAIRS OF 1's AND 0's.

TABLE 2.1-1 FOURIER SERIES FOR FILTERED SIGNAL QUADRATURE COMPONENTS

| Bit patterns | u(t) | v(t) | |
|---|---|---|---|
| $1\underline{1}1, 0\underline{0}0$ | $\pm a_0 \sin(\pi h t/T)$ | $a_0 \cos(\pi h t/T)$ | $\tan\phi(t) = \frac{u(t)}{v(t)}$ |
| $1\underline{0}0(1), 0\underline{1}1(0)$ | $\mp [c_4 \sin(\pi t/2T) - c_5 \sin(3\pi t/2T)]$ | $c_6 + c_7 \cos(\pi t/T) - c_8 \cos(2\pi t/T)$ | |
| $0\underline{1}0(1), 1\underline{0}1(0)$ | $\pm c_1 \cos(\pi t/T)$ | $c_2 - c_3 \cos(2\pi t/T)$ | |
| $1\underline{1}0(0), 0\underline{0}1(1)$ | $\pm [c_4 \cos(\pi t/2T) + c_5 \cos(3\pi t/2T)]$ | $c_6 - c_7 \cos(\pi t/T) - c_8 \cos(2\pi t/T)$ | |

<u>Coefficients</u>:

$h = 2f_d T$, $D = W_{IF} T = 2B_0 T$

$a_0 = |H_0(h/2T)| = e^{-\pi h^2/8D^2}$

$a_1 = |H_0(1/2T)| = e^{-\pi/8D^2}$

$a_2 = |H_0(1/T)| = e^{-\pi/2D^2}$

$a_3 = |H_0(1/4T)| = e^{-\pi/32D^2}$

$a_4 = |H_0(3/4T)| = e^{-9\pi/32D^2}$

$c_1 = a_1 \cdot 4h \cos(\pi h/2)/\pi(1-h^2)$

$c_2 = 2 \sin(\pi h/2)/\pi h$, $c_6 = \sin(\pi h)/\pi h$

$c_3 = a_2 \cdot 4h \sin(\pi h/2)/\pi(4-h^2)$

$c_4 = a_3 \cdot 8h \cos(\pi h)/\pi(1-4h^2)$

$c_5 = a_4 \cdot 8h \cos(\pi h)/\pi(9-4h^2)$

$c_7 = c_6 \cdot a_1 \cdot 2h^2/(1-h^2)$

$c_8 = c_6 \cdot a_2 \cdot 2h^2/(4-h^2)$

TABLE 2.1-2

EXAMPLE VALUES FOR SIGNAL PARAMETERS DESCRIBED IN TABLE 2.1-1

Case: $h = 0.7$, $D = W_{IF}T = 1.0$, Gaussian filter

| Parameter | Value |
|---|---|
| $a_0$ | 0.82496 |
| $a_1$ | 0.67523 |
| $a_2$ | 0.20788 |
| $a_3$ | 0.90649 |
| $a_4$ | 0.41330 |
| $c_1$ | $0.79339\ a_1 = 0.53572$ |
| $c_2$ | 0.81033 |
| $c_3$ | $0.22625\ a_2 = 0.047032$ |
| $c_4$ | $1.0914\ a_3 = 0.98935$ |
| $c_5$ | $-0.14883\ a_4 = -0.061511$ |
| $c_6$ | 0.36788 |
| $c_7$ | $0.70691\ a_1 = 0.47732$ |
| $c_8$ | $0.10271\ a_2 = 0.021352$ |

### 2.1.3 Limiter Output and Total Phase

A simplified receiver block diagram was given previously as Figure 2.1-1. As shown in Figure 2.1-4 the digital FM demodulator consists of a limiter followed by a discriminator and an integrate-and-dump filter, whose output is sampled to yield the demodulated data sequence.

The purpose of the limiter, assumed to be an ideal bandpass limiter, is to remove any amplitude modulation on the I.F. signal. Its output, y(t), is a constant amplitude sinusoid with the same total phase (including modulation) as the I.F. waveform:

$$y(t) = \text{constant} \cdot \cos\{\omega_0 t + \text{Phase}\,[x(t)]\}. \tag{2.1-25}$$

The total phase $\phi(t)$ of x(t) is found from the following development:

$$\begin{aligned} x(t) &= [a(t)A\cos\theta(t) + n_c(t)]\cos(\omega_0 t+\theta_0) \\ &\quad - [a(t)A\sin\theta(t) + n_s(t)]\sin(\omega_0 t+\theta_0) \\ &= \text{Env}[x(t)] \cdot \cos[\omega_0 t + \theta_0 + \theta(t) + \eta(t)], \end{aligned} \tag{2.1-26a}$$

where $\eta(t)$ is a phase noise term, and

$$\begin{aligned} \tan[\theta(t) + \eta(t)] &\equiv \tan\phi(t) \\ &= \frac{a(t)A\sin\theta(t) + n_s(t)}{a(t)A\cos\theta(t) + n_c(t)} = \frac{u(t) + n_s(t)}{v(t) + n_c(t)} . \end{aligned} \tag{2.1-26b}$$

This development uses the Rician decomposition of the bandpass noise, referenced to the carrier frequency and phase:

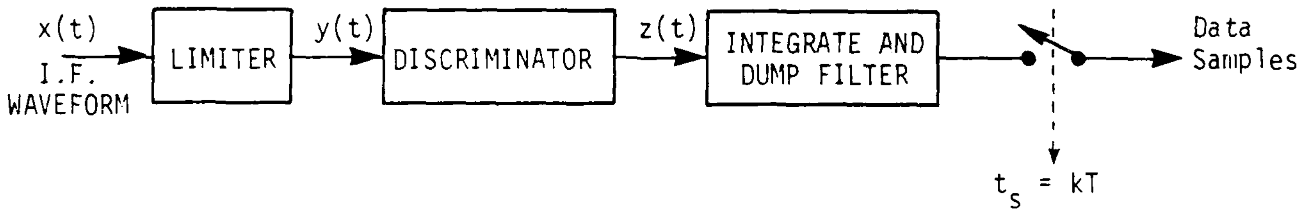



FIGURE 2.1-4 DETAILS OF DIGITAL FM DEMODULATOR

$$n(t) = n_c(t)\cos(\omega_0 t+\theta_0) - n_s(t)\sin(\omega_0 t+\theta_0). \tag{2.1-26c}$$

Recall that this noise may include jamming noise. At any instant, the quadrature components $n_c(t)$ and $n_s(t)$ are independent zero-mean Gaussian random variables with equal variances

$$\sigma^2 = \begin{cases} N_0 W_{IF}, & \text{hop not jammed} \\ (N_0+N_{0J})W_{IF}, & \text{hop jammed.} \end{cases} \tag{2.1-27}$$

Note that we can write the phase noise term as

$$\eta(t) = \tan^{-1}\left[\frac{\nu_s(t)\cos\phi(t) - \nu_c(t)\sin\phi(t)}{\sqrt{2\rho(t)} + \nu_s(t)\sin\phi(t) + \nu_c(t)\cos\phi(t)}\right] \tag{2.1-28a}$$

in which $\nu_s(t)$ and $\nu_c(t)$ are

$$\nu_s(t) \triangleq n_s(t)/\sigma, \quad \nu_c(t) \triangleq n_c(t)/\sigma, \tag{2.1-28b}$$

unit-variance Gaussian random variables, and

$$\rho(t) \triangleq a^2(t)A^2/2\sigma^2 = [u^2(t)+v^2(t)]/2\sigma^2 \tag{2.1-28c}$$

is the SNR (carrier-to-noise ratio) with time variation due to the I.F. filter-induced amplitude modulation. A further simplification results from using a rotational transformation of noise variables to write

$$\eta(t) = \tan^{-1}\left[\frac{\nu_s'(t)}{\sqrt{2\rho(t)} + \nu_c'(t)}\right]; \tag{2.1-29a}$$

we recognize that since

$$(\nu_s')^2 + (\nu_c')^2$$

$$= (\nu_s \cos\phi - \nu_c \sin\phi)^2 + (\nu_s \sin\phi + \nu_c \cos\phi)^2$$

$$= \nu_s^2 + \nu_c^2 , \qquad (2.1\text{-}29b)$$

the distribution of the rotated noise components $\nu_c'$ and $\nu_c'$ is identical to that of the actual components $\nu_c$ and $\nu_s$. Thus at a given time instant, the phase noise term $\eta(t)$ is independent of the signal phase and additive to it. It is easily shown that the probability density function (pdf) for $\eta(t)$ is (see [4], chapter 9, e.g.)

$$p_\eta(\alpha) = \frac{e^{-\rho}}{2\pi} \int_0^\infty dx \; x \; e^{-(x^2 - 2\sqrt{2\rho}x \cos\alpha)/2} \qquad (2.1\text{-}30a)$$

$$= \frac{e^{-\rho}}{2\pi} + \sqrt{\frac{\rho}{\pi}} \; e^{-\rho \sin^2\alpha} \cos\alpha \; Q(-\sqrt{2\rho} \cos\alpha), \; |\alpha| < \pi ; \qquad (2.1\text{-}30b)$$

and therefore the total phase pdf is

$$p_\phi(\alpha) = p_\eta(\alpha - \phi)$$

$$= \frac{e^{-\rho}}{2\pi} + \sqrt{\frac{\rho}{\pi}} \; e^{-\rho \sin^2(\alpha - \phi)} \cos(\alpha - \phi) \; Q[-\sqrt{2\rho} \cos(\alpha - \phi)], \; |\alpha - \phi| < \pi. \qquad (2.1\text{-}30c)$$

In these expressions, $Q(\cdot)$ is defined as

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty d\alpha \; e^{-\alpha^2/2}$$

$$= \frac{1}{2} \operatorname{erfc}(x/\sqrt{2}). \qquad (2.1\text{-}30d)$$

### 2.1.4 Discriminator and Baseband Outputs

The discriminator output extracts $2\pi$ times the instantaneous frequency deviation from the carrier $f_0$. This quantity is

$$z(t) = 2\pi[f(t)-f_0] = \dot{\phi}(t) + \dot{\eta}(t) = \dot{\Phi}(t). \tag{2.1-31a}$$

The baseband filter, assumed to be of the integrate-and-dump type, operates on $z(t)$ to produce the differential phase at sample time $t_s$:

$$\Delta\Phi(t_s) = \int_{t_s-T}^{t_s} d\tau\, z(\tau) = \phi(t_s) - \phi(t_s-T) + \eta(t_s) - \eta(t_s-T)$$

$$= \Delta\phi(t_s) + \Delta\eta(t_s). \tag{2.1-31b}$$

Although we have written the total differential phase $\Delta\Phi$ as the sum of a signal differential phase term and a noise differential phase term, in general the "differential phase noise" term $\Delta\eta$ is not additive and independent of $\Delta\phi$, but rather depends upon $\Delta\phi$. Further discussion of phase noise, including "FM clicks", is given in Sections 2.1.5 and 2.1.6.

Without noise, the differential phase output of the digital FM demodulator is

$$\Delta\phi = \tan^{-1}\left[\frac{u(t_s)}{v(t_s)}\right] - \tan^{-1}\left[\frac{u(t_s-T)}{v(t_s-T)}\right], \tag{2.1-32a}$$

where the principal values of the arctangents can be used if the trajectory of $\phi(t) = \tan^{-1}[u/v]$ is such that $|\Delta\phi| < \pi$. Otherwise, there is an inherent $2\pi$-radian ambiguity in the arctangent. For the signal only, these requirements are satisfied, and the resolution of arctangent ambiguities can be successfully accomplished by rewriting (2.1-31a), using the relation

$$\theta_1 - \theta_2 = \tan^{-1}\{\tan(\theta_1-\theta_2)\}$$

$$= \tan^{-1}\left\{\frac{\tan\theta_1 - \tan\theta_2}{1 + \tan\theta_1\tan\theta_2}\right\} ,$$

and identifying $\theta_1$ and $\theta_2$ to be the first and second terms, respectively of (2.1-32a), it is straightforward to find that

$$\Delta\phi = \tan^{-1}\left[\frac{u(t_s)v(t_s-T) - u(t_s-T)v(t_s)}{u(t_s)u(t_s-T) + v(t_s)v(t_s-T)}\right]. \qquad (2.1\text{-}32b)$$

A plot of $\Delta\phi(t)$ for the various data sequences, commonly called an "eye pattern", is shown in Figure 2.1-5 for h = 0.7 and D = 1.0. Note the "closing" of the "eye" for the alternating bit sequences, due to the filtering in the receiver, and the dependence of the $\Delta\phi$ values at the bit times upon the sequence.

The value of $\Delta\phi(t)$ at t = 0 represents the sampled data value as recovered by the demodulator. From Table 2.1-1 we can calculate the sample amplitudes observed in Figure 2.1-5 in terms of the patterns and filter parameters. These are summarized in Table 2.1-3. From the figure and table we observe the effect that the I.F. filter has on the data output. Ideally, the differential phase is $\pm\pi h$, as it is for the all-one's or all-zeros patterns. For the other patterns, the phase distortion causes intersymbol interference, as the values of adjacent bits affect that of the current one, with the worst-case being alternating one's and zeros. However, as Table 2.1-3 demonstrates, the average of the time-varying SNR is higher for the alternating patterns, since the instantaneous frequency for these patterns is, on the average, nearer to the filter center frequency than is that for the non-alternating patterns.

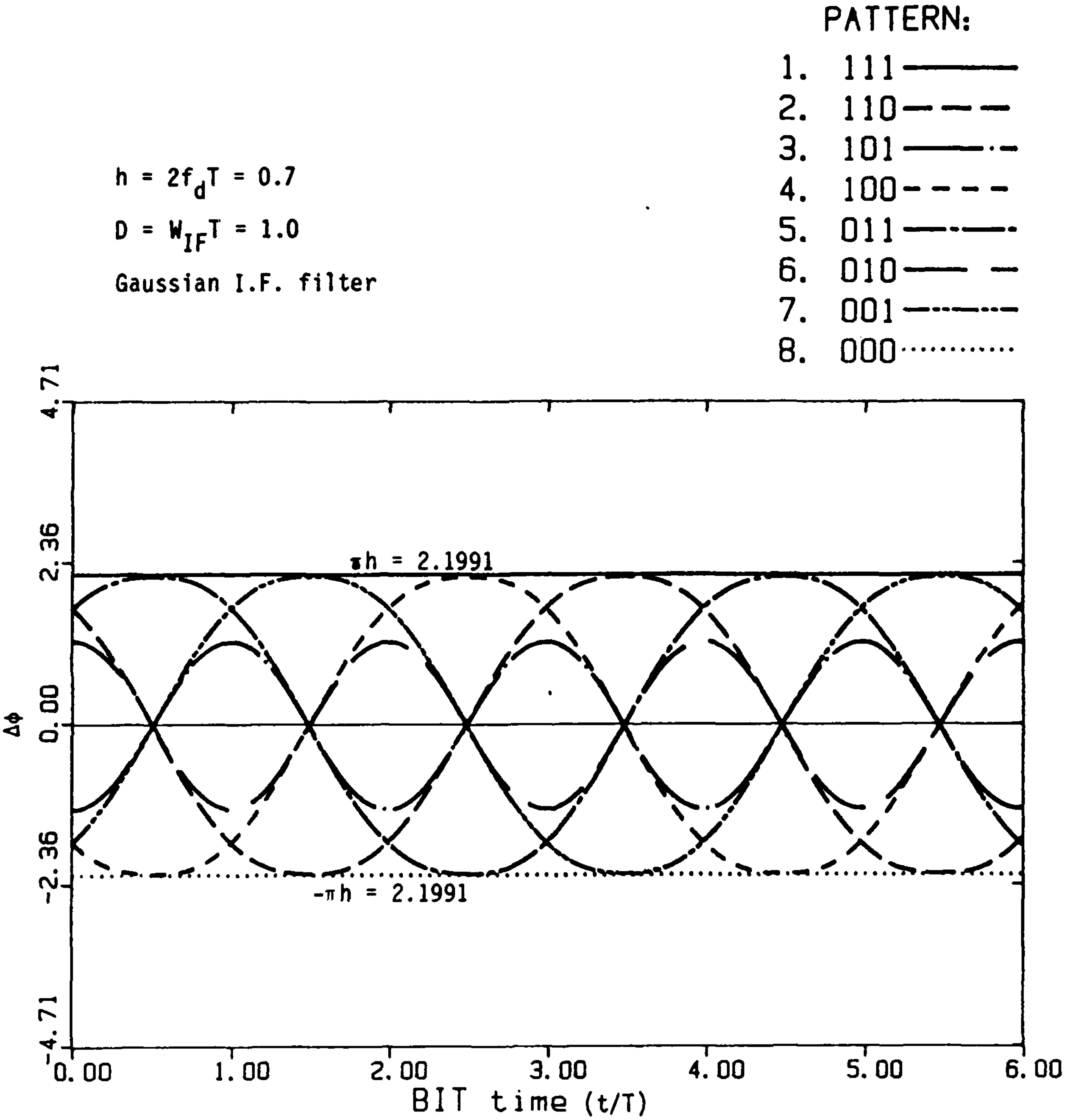



FIGURE 2.1-5 DATA EYE PATTERNS (RUNNING VALUE OF INTEGRATE-AND-DUMP FILTER OUTPUT) WITHOUT NOISE OR JAMMING

TABLE 2.1-3 EXAMPLE DATA EYE PATTERN AMPLITUDES AND SNR PARAMETERS

Case: $h = 0.7$, $D = W_{IF}T = 1.0$, Gaussian-shaped I.F. filter

SNR parameters: $U \triangleq [\rho(0) + \rho(-T)]/2$ CNR = carrier-to-noise power ratio

$V \triangleq [\rho(0) - \rho(-T)]/2$

$W \triangleq \sqrt{U^2 - V^2} = \sqrt{\rho(0)\rho(-T)}$

| Data Pattern | $\rho(0)$/CNR | $\rho(-T)$/CNR | $\Delta\phi$(rad) | U/CNR | V/CNR | W/CNR |
|---|---|---|---|---|---|---|
| 111, 000 | .6806 | .6806 | ± 2.1991 | .6806 | 0.0 | .6806 |
| 011, 100 | .6787 | .8780 | ± 1.7108 | .7784 | -.0997 | .7719 |
| 010, 101 | .8696 | .8696 | ± 1.2239 | .8696 | 0.0 | .8696 |
| 110, 001 | .8780 | .6787 | ± 1.7108 | .7784 | .0997 | .7719 |
| Algebraic expressions: | | | | | | |
| 111, 000 | $a_0^2$ | $a_0^2$ | $\pm \pi h$ | $a_0^2$ | 0 | $a_0^2$ |
| 011, 100 | (U+V)/CNR | (U-V)/CNR | $\pm \tan^{-1}\left(\frac{c_4+c_5}{c_6-c_7-c_8}\right)$ | $\frac{1}{2}(c_4+c_5)^2 + c_7^2+(c_6-c_8)^2$ | $2c_7(c_6-c_8) - \frac{1}{2}(c_4+c_5)^2$ | $\sqrt{(U^2-V^2)}$/CNR |
| 010, 101 | " | " | $\pm 2\tan^{-1}\left(\frac{c_1}{c_2-c_3}\right)$ | $c_1^2 + (c_2-c_3)^2$ | 0 | " |
| 110, 001 | " | " | $\pm \tan^{-1}\left(\frac{c_4+c_5}{c_6-c_7-c_8}\right)$ | $\frac{1}{2}(c_4+c_5)^2 + c_7^2+(c_6-c_8)^2$ | $-2c_7(c_6-c_8) + \frac{1}{2}(c_4+c_5)^2$ | " |

### 2.1.5 Differential Phase Distribution

As we have seen, the digital FM receiver output produces the differential phase $\Delta\Phi(t)$, where the difference is between the two values of the signal-plus-bandpass noise total phase occurring at the sampling time and one bit period earlier. The probability distribution of the differential phase $\Delta\Phi$ is found by starting with the joint distribution of the additive noise components. Let the noise quadrature components be given by

$$n_c(t) \equiv n_1 \qquad n_c(t-T) \equiv n_3$$

$$n_s(t) \equiv n_2 \qquad n_s(t-T) \equiv n_4 \ . \tag{2.1-33a}$$

If the autocorrelation function for the bandpass noise is given by

$$R_n(\tau) = \sigma^2[r(\tau)\cos\omega_0\tau - \lambda(\tau)\sin\omega_0\tau], \tag{2.1-33b}$$

then the column vector $\underline{n} = (n_1, n_2, n_3, n_4)^T$ is a zero-mean multivariate Gaussian random vector with covariance matrix

$$\Sigma = \sigma^2 \begin{bmatrix} 1 & 0 & r & \lambda \\ 0 & 1 & -\lambda & r \\ r & -\lambda & 1 & 0 \\ \lambda & r & 0 & 1 \end{bmatrix} \equiv \sigma^2 C, \tag{2.1-33c}$$

where

$$r \equiv r(T) \text{ and } \lambda \equiv \lambda(T).$$

For a flat noise spectrum going into the receiver I.F. filter, the autocorrelation function $R_n(\tau)$ is determined by the shape of the filter. For example, if the filter frequency characteristic (passband) is symmetric about the signal center frequency $f_0$, then $\lambda(\tau)$ is identically zero. Using the Gaussian-shaped filter introduced previously in equation (2.1-11), the noise spectrum is

$$\frac{N_0}{2}\left[e^{-\pi(f-f_0)^2/W_{IF}^2} + e^{-\pi(f+f_0)^2/W_{IF}^2}\right], \tag{2.1-34a}$$

for which the autocorrelation function is

$$R_n(\tau) = N_0 W_{IF}\, e^{-\pi(W_{IF}\tau)^2} \cos\omega_0\tau, \tag{2.1-34b}$$

so that for this filter

$$r = e^{-\pi D^2},\quad \lambda = 0,\quad \sigma^2 = N_0 W_{IF} \tag{2.1-34c}$$

in (2.1-33b), using $D = W_{IF}T$.

The multivariate probability density function for the noise vector $\underline{n}$ is

$$p_{\underline{n}}(\underline{\alpha}) = \left(4\pi^2\sqrt{\det \Sigma}\right)^{-1} \exp\left\{-\frac{1}{2}\,\underline{\alpha}^T \Sigma^{-1} \underline{\alpha}\right\}. \tag{2.1-35a}$$

If we define the signal-plus-noise vector as $\underline{x}$, with signal component $\underline{x}_0$ given by

$$\underline{x}_0 = \begin{bmatrix} Aa_1 \cos\phi_1 \\ Aa_1 \sin\phi_1 \\ Aa_2 \cos\phi_2 \\ Aa_2 \sin\phi_2 \end{bmatrix} \tag{2.1-35b}$$

then the pdf for $\underline{x}$ is

$$p_{\underline{x}}(\underline{\alpha}) = p_{\underline{n}}(\underline{\alpha}-\underline{x}_0); \tag{2.1-35c}$$

for convenience we can normalize each noise component by $\sigma$ to get unit-variance random variables, with the normalized signal component vector (mean) then expressible as

$$\underline{s} \doteq \underline{x}_0/\sigma = \begin{bmatrix} \sqrt{2\rho_1}\cos\phi_1 \\ \sqrt{2\rho_1}\sin\phi_1 \\ \sqrt{2\rho_2}\cos\phi_2 \\ \sqrt{2\rho_2}\sin\phi_2 \end{bmatrix} \tag{2.1-35d}$$

with $\rho_1$ and $\rho_2$ being the SNR at times t and t-T, respectively. The normalized pdf then is

$$p_{\underline{x}}(\underline{\alpha}) = (4\pi^2\sqrt{\det C})^{-1} \exp\left\{ -\frac{1}{2}(\underline{\alpha}-\underline{s})^T C^{-1}(\underline{\alpha}-\underline{s}) \right\} \tag{2.1-36a}$$

$$\begin{aligned} &= [4\pi^2(1-r^2-\lambda^2)]^{-1} \\ &\times \exp\left\{ -\frac{1}{2}\left[\frac{1}{1-r^2-\lambda^2}\right] \left[(\alpha_1-\sqrt{2\rho_1}\cos\phi_1)^2 + (\alpha_2-\sqrt{2\rho_1}\sin\phi_1)^2\right.\right. \\ &\quad -2r(\alpha_1-\sqrt{2\rho_1}\cos\phi_1)(\alpha_3-\sqrt{2\rho_2}\cos\phi_2) \\ &\quad -2\lambda(\alpha_1-\sqrt{2\rho_1}\cos\phi_1)(\alpha_4-\sqrt{2\rho_2}\sin\phi_2) \\ &\quad +2\lambda(\alpha_2-\sqrt{2\rho_1}\sin\phi_1)(\alpha_3-\sqrt{2\rho_2}\cos\phi_2) \\ &\quad -2r(\alpha_2-\sqrt{2\rho_1}\sin\phi_1)(\alpha_4-\sqrt{2\rho_2}\sin\phi_2) \\ &\quad \left.\left. +(\alpha_3-\sqrt{2\rho_2}\cos\phi_2)^2 + (\alpha_4-\sqrt{2\rho_2}\sin\phi_2)^2\right]\right\} . \end{aligned} \tag{2.1-36b}$$

Since we are interested in the phases at the sample times, it is appropriate to change from rectangular to polar coordinates using the transformation

$$\left.\begin{aligned} R_1 \cos\Phi_1 &= \alpha_1 \\ R_1 \sin\Phi_1 &= \alpha_2 \\ R_2 \cos\Phi_2 &= \alpha_3 \\ R_2 \sin\Phi_2 &= \alpha_4 \end{aligned}\right\} \quad \left\{\begin{aligned} & R_1 \geq 0 \\ & |\Phi_1-\phi_1| \leq \pi \\ & R_2 \geq 0 \\ & |\Phi_2-\phi_2| \leq \pi. \end{aligned}\right. \tag{2.1-37a}$$

If this transformation is employed, we obtain the joint pdf of envelopes and phases given by

$$p_{EP}(R_1, R_2, \Phi_1, \Phi_2) = R_1R_2p_{\underline{X}}(R_1\cos\Phi_1, R_1 \sin\Phi_1, R_2 \cos\Phi_2, R_2 \sin\Phi_2)$$

$$= \frac{R_1 R_2}{4\pi^2(1-\mu^2)} \exp\left\{ \frac{-1}{1-\mu^2} \left[(R_1^2+R_2^2)/2 + \rho_1 + \rho_2\right.\right.$$

$$-\mu R_1R_2 \cos(\Phi_1-\Phi_2+\xi) - XR_1 \cos(\Phi_1-v)$$

$$\left.\left.-Y R_2 \cos(\Phi_2-w) - 2\mu\sqrt{\rho_1\rho_2} \cos(\phi_1-\phi_2+\xi)\right]\right\} \tag{2.1-37b}$$

in which we use the notations

$$\mu^2 = r^2+\lambda^2, \quad \xi = \tan^{-1}(\lambda/r) \tag{2.1-37c}$$

$$X^2 = 2\rho_1+2\mu^2\rho_2 - 4\mu\sqrt{\rho_1\rho_2} \cos(\phi_1-\phi_2+\xi) \tag{2.1-37d}$$

$$\tan v = [\sqrt{2\rho_1} \sin\phi_1-\mu\sqrt{2\rho_2} \sin(\phi_2-\xi)]/[\sqrt{2\rho_1} \cos\phi_1-\mu\sqrt{2\rho_2} \cos(\phi_2-\xi)] \tag{2.1-37e}$$

$$\gamma^2 = 2\rho_2 + 2\mu^2\rho_1 - 4\mu\sqrt{\rho_1\rho_2}\,\cos(\phi_1-\phi_2+\xi) \qquad (2.1\text{-}37f)$$

and

$$\tan w = [\sqrt{2\rho_2}\,\sin\phi_2-\mu\sqrt{2\rho_1}\,\sin(\phi_1+\xi)]/[\sqrt{2\rho_2}\,\cos\phi_2-\mu\sqrt{2\rho_1}\,\cos(\phi_1+\xi)]. \qquad (2.1\text{-}37g)$$

As a check, we note that for $\mu = 0$ (no noise correlation), the pdf in (2.1-37) reduces to the product of the pdf's for each sample time:

$$p_{EP}(R_1,\, R_2\,, \Phi_1\,, \Phi_2\,;\, \mu=0)$$
$$= p(R_1\,, \Phi_1\,)\; p(R_2\,, \Phi_2\,) \qquad (2.1\text{-}38a)$$

where each pdf is of the form

$$p(R,\Phi) = \frac{R}{2\pi}\; \exp\left\{ -\frac{R^2}{2} - \rho + R\sqrt{2\rho}\,\cos(\Phi - \phi)\right\}, \qquad (2.1\text{-}38b)$$

as discussed in connection with equation (2.1-30).

Having expressed the joint pdf for the envelopes and phases in (2.1-37), the general procedure is to integrate over the envelope variables to give a phase-only pdf,

$$p_\Phi(\Phi_1\,, \Phi_2) = \int_0^\infty dR_1 \int_0^\infty dR_2\; p_{EP}(R_1,\, R_2,\, \Phi_1\,, \Phi_2\,), \qquad (2.1\text{-}39)$$

and then to find the pdf for the differential phase $\Delta\Phi = \Phi_1 - \Phi_2$. Various approaches have been used for carrying out the integrations shown in (2.1-39). Due to the complexity of the expression, the relative virtue of any particular approach lies in the feasibility of computing the phase pdf's obtained by the approach. Here we will summarize some approaches that have been tried, and present the corresponding results.

2.1.5.1 Fourier Series Approach

What we call the "Fourier Series approach" is one used by Middleton [4, chapter 9]. The procedure is to utilize the Fourier series expansion

$$e^{a \cos b} = \sum_{k=0}^{\infty} \varepsilon_k I_k(a) \cos kb, \tag{2.1-40a}$$

in which

$$\varepsilon_k = \begin{cases} 1, & k = 0 \\ 2, & k > 0 \end{cases}, \tag{2.1-40b}$$

and $I_k(\cdot)$ is the modified Bessel function of the first kind, of order k. Using this expansion on the $\cos(\Phi_1 - v) = \cos(\Phi_2 + \Delta\Phi - v)$ and $\cos(\Phi_2 - w)$ terms in the exponent of (2.1-37b), and integrating the resulting product of series over $\Phi_2$ (a $2\pi$ interval) yields a single series:

$$\int_{(2\pi)} d\Phi_2 \exp\left\{\frac{XR_1}{1-\mu^2}\cos(\Phi_2 + \Delta\Phi - v) + \frac{YR_2}{1-\mu^2}\cos(\Phi_2 - w)\right\}$$

$$= 2\pi \sum_{k=0}^{\infty} \varepsilon_k I_k\left(\frac{XR_1}{1-\mu^2}\right) I_k\left(\frac{YR_2}{1-\mu^2}\right) \cos k(\Delta\Phi - v + w). \tag{2.1-41}$$

Next in the procedure is to expand the $R_1 R_2$ term in the exponent to obtain

$$\exp\left\{\frac{R_1 R_2}{1-\mu^2}\cos(\Delta\Phi - \xi)\right\}$$

$$= \sum_{\ell=0}^{\infty} \varepsilon_\ell I_\ell\left[\frac{R_1 R_2}{1-\mu^2}\right] \cos \ell(\Delta\Phi - \xi)$$

$$= \sum_{\ell=0}^{\infty} \sum_{m=0}^{\infty} \varepsilon_\ell \left[\frac{\mu R_1 R_2}{2(1-\mu^2)}\right]^{2m+\ell} \frac{1}{m!\,(m+\ell)!} \cos \ell(\Delta\Phi - \xi), \tag{2.1-42}$$

in which a series expansion for the Bessel function has been used. At this point, the integrations over $R_1$ and $R_2$ can be carried out. A synthesis of formulas 6.643.2 and 9.220 in [5] provides the integration formula needed:

$$\int_0^\infty dR\, R^{1+2m+\ell}\, e^{-\alpha R^2/2}\, I_k(\beta R\sqrt{2})$$

$$= \int_0^\infty dx\, x^{m+\ell/2}\, e^{-\alpha x}\, I_k(2\beta\sqrt{x}) \cdot 2^{m+\ell/2}$$

$$= \frac{\Gamma[m+1+(k+\ell)/2]}{k!}\, \beta^k\, \alpha^{-m-1-(k+\ell)/2}\, 2^{m+\ell/2}$$

$$\times\ {}_1F_1[m+1+(k+\ell)/2;\ k+1;\ \beta^2/\alpha], \qquad (2.1\text{-}43)$$

where $\Gamma(\cdot)$ is the gamma function and ${}_1F_1(a;b;c)$ is the confluent hypergeometric function. After using this formula twice, the pdf for $\Delta\phi$ is a triple infinite series of the Fourier type:

$$p_{\Delta\phi}(x) = \sum_{k=0}^{\infty} \sum_{\ell=0}^{\infty} \sum_{m=0}^{\infty} K(k,\ell,m)\, \cos k(\Delta\phi - v + w)\, \cos\ell(\Delta\phi - \xi), \qquad (2.1\text{-}44a)$$

where the coefficients are given by

$$K(k,\ell,m) = \frac{1}{2\pi(1-\mu^2)} \exp\left\{-\frac{\rho_1+\rho_2-\mu\sqrt{\rho_1\rho_2}\cos(\Delta\phi+\xi)}{1-\mu^2}\right\}$$

$$\times \frac{\varepsilon_k\,\varepsilon_\ell}{m!\,(m+\ell)!\,k!\,k!}\, \Gamma^2[m+1+(k+\ell)/2]$$

$$\times \left(\frac{XY}{2}\right)^k (1-\mu^2)^{2-k}\, \mu^{2m+\ell}$$

$$\times\ {}_1F_1[m+1+(k+\ell)/2;\ k+1;\ X^2/2(1-\mu^2)]$$

$$\times\ {}_1F_1[m+1+(k+\ell)/2;\ k+1;\ Y^2/2(1-\mu^2)]. \qquad (2.1\text{-}44b)$$

As a special case, note that for no correlation ($\mu$=0), the pdf reduces to the m=$\ell$=0 case of a single series:

$$p_{\Delta\phi}(x) = \frac{e^{-\rho_1-\rho_2}}{2} \sum_{k=0}^{\infty} \varepsilon_k \frac{\Gamma^2(1+k/2)}{(k!)^2} (\rho_1\rho_2)^{k/2} \cos(\Delta\Phi - \Delta\phi)$$

$$\times \; {}_1F_1(1+k/2;k+1;\rho_1)\,{}_1F_1(1+k/2;k+1;\rho_2). \tag{2.1-45}$$

This special case has been obtained by Mizuno et al [6], and is recognizable as the self-convolution of Middleton's series form [4] for the single phase pdf (2.1-30), often cited in the communications literature.

2.1.5.2 <u>2$\pi$ Modularity Issues</u>

The procedure used in deriving the differential phase pdf (2.1-44) involved a subtle but very significant assumption. In (2.1-41), integration over $\Phi_2$ was assumed to be performed over a 2$\pi$ interval. As presented, this integration preceded integration over the envelope variables, but the order of integration is not an issue. What is important is that this step is not strictly correct unless the resulting differential phase is interpreted as a modulo 2$\pi$ quantity.

To prove the preceding assertion, consider the following reasoning. First, we note that the mathematical expression for the joint pdf $p_\Phi(\Phi_1,\Phi_2)$ is periodic, that is,

$$f_\Phi(\Phi_1 \pm 2n\pi, \Phi_2 \pm 2m\pi) = f_\Phi(\Phi_1, \Phi_2), \tag{2.1-46}$$

where $f_\Phi$ is the expression. Next, we state that the joint pdf consists of the mathematical expression plus the restriction of $\Phi_1$ and $\Phi_2$ to some principal interval (otherwise, the probability "mass" under the pdf would be infinite). Since for no noise, $\Phi_1 \to \phi_1$ and $\Phi_2 \to \phi_2$, it is reasonable to state that

$$p_\Phi(\phi_1, \phi_2) = \begin{cases} f_\Phi(\Phi_1, \Phi_2), |\Phi_1 - \phi_1| < \pi \text{ and } |\Phi_2 - \phi_2| < \pi \\ 0, \text{ otherwise.} \end{cases} \tag{2.1-47}$$

Now, the formal procedure for obtaining the pdf of $\Delta\Phi = \Phi_1 - \Phi_2$ consists of performing the integral

$$p_{\Delta\Phi}(y) = \int_L^U dx\, p_\Phi(x+y, x) \ , \ |\Delta\Phi - \Delta\phi| < 2\pi, \tag{2.1-48a}$$

where the limits of integration are, by (2.1-47),

$$U = \min[\phi_2 + \pi, \phi_2 + \pi - (\Delta\Phi - \Delta\phi)] \tag{2.1-48b}$$

and

$$L = \max[\phi_2 - \pi, \phi_2 - \pi + (\Delta\Phi - \Delta\phi)]. \tag{2.1-48c}$$

This conventional procedure yields a pdf $p_{\Delta\Phi}(\cdot)$ which is nonzero on a $4\pi$ interval. For example, for noise only and no correlation, the "actual" differential phase pdf is triangular:

$$p_{\Delta\Phi}(y; A=\mu=0) = \begin{cases} (2\pi - |\Delta\Phi|)/4\pi^2 \ , \ |\Delta\Phi| < 2\pi; \\ 0, \text{ otherwise.} \end{cases} \tag{2.1-49}$$

But we can also speak of a "modulo $2\pi$" differential phase, defined as

$$\psi \triangleq (\Delta\Phi - \Delta\phi) \bmod 2\pi + \Delta\phi \tag{2.1-50a}$$

$$= \begin{cases} \Delta\Phi \ , \ |\Delta\Phi - \Delta\phi| < 2\pi \\ \Delta\Phi - 2\pi, \ \ 2\pi < \Delta\Phi - \Delta\phi < 4\pi \\ \Delta\Phi + 2\pi \ , \ \ -4\pi < \Delta\Phi - \Delta\phi < -2\pi. \end{cases} \tag{2.1-50b}$$

The pdf for this modulo $2\pi$ differential phase is([7], [8]) the aliased "actual" pdf:

$$p_{\psi}(y) = \begin{cases} \sum_{k=-\infty}^{\infty} p_{\Delta\phi}(y+2k\pi), & |y-\Delta\phi| < \pi ; \\ 0 , & \text{otherwise.} \end{cases} \tag{2.1-50c}$$

Again using the noise-only example, modulo $2\pi$ the pdf for the differential phase is uniform:

$$p_{\psi}(y;A=\mu=0) = \frac{1}{2\pi} , \quad |\psi| < \pi . \tag{2.1-51}$$

With these considerations in mind, we can show the relationship of the modulo $2\pi$ differential phase pdf to the integral form of the "actual" $\Delta\phi$ pdf. Using (2.1-50c) and (2.1-48), we find that

$$p_{\psi}(y) = \int_{L_{-1}}^{U_{-1}} dx\ f_{\phi}(x+y-2\pi,x) + \int_{L_0}^{U_0} dx\ f_{\phi}(x+y,\ x) + \int_{L_1}^{U_1} dx\ f_{\phi}(x+y+2\pi\ ,\ x), \tag{2.1-52a}$$

where

$$U_k = \min[\phi_2+\pi,\ \phi_2+\pi-(y-\Delta\phi+2k\pi)] \tag{2.1-52b}$$

and

$$L_k = \max[\phi_2-\pi,\ \phi_2-\pi+(\Delta\phi-y-2k\pi)]. \tag{2.1-52c}$$

For $y(=\Delta\phi) < \Delta\phi$ we have $U_{-1} < L_{-1}$, so that the first integral in (2.1-52a) is zero, and

$$U_0 = \phi_2+\pi \quad , L_1 = \phi_2-\pi \tag{2.1-53a}$$

and

$$L_0 = U_1 = \phi_2-\pi + \Delta\phi-\Delta\Phi. \tag{2.1-53b}$$

Since $f_\zeta(\cdot)$ is periodic, we therefore have

$$p_\zeta(y) = \int_{\phi_2-\pi}^{\phi_2+\pi} dx\ f_\zeta(x+y, x), \ |y-\Delta\phi| < \pi , \tag{2.1-54}$$

with the same result for $y > \Delta\phi$.

Our conclusion therefore is that integration over a $2\pi$ interval of $\Phi_2$ in the derivation of the differential phase pdf produces a result that pertains not to the actual differential phase observed at the digital FM demodulator output, but to a modulo $2\pi$ version of it.

The question is, which version of the differential phase is appropriate for analyzing digital FM performance? Not using $2\pi$ modularity greatly complicates the analysis and computation of the pdf (see [9] and [10], for example). However, it is incorrect to say that the voltage at the output of the receiver is proportional to a modulo $2\pi$ differential phase, because there is no mechanism in the discriminator or in the integrate-and-dump filter which would induce this modularity. For high SNR, the choice is somewhat arbitrary, since the unaliased pdf is negligible for $|\Delta\Phi-\Delta\phi| > \pi$. On this account, we shall use modulo $2\pi$ expressions unless stated otherwise.

### 2.1.5.3 Characteristic Function Approach

Using a modified characteristic function approach, Pawula, Rice, and Roberts [8] have developed expressions for the modulo $2\pi$ differential phase distributions. The pertinent results are the following:

$$p_\psi(y) = \frac{1-\mu^2}{4\pi} \int_{-\pi/2}^{\pi/2} dx \, \frac{e^{-E(x)} \cos x}{[1-(r\cos y + \lambda \sin y)\cos x]^2} \left[ 1 - E(x) + 2\, \frac{U - W(r\cos\Delta\phi + \lambda \sin\Delta\phi)}{1-\mu^2} \right] \quad (2.1\text{-}55a)$$

where

$$E(x) = \frac{U - V\sin x - W\cos(\Delta\phi - y)\cos x}{1-(r\cos y + \lambda \sin y)\cos x} \quad (2.1\text{-}55b)$$

and

$$U \triangleq (\rho_1 + \rho_2)/2 \quad (2.1\text{-}55c)$$

$$V \triangleq (\rho_1 - \rho_2)/2 \quad (2.1\text{-}55d)$$

$$W \triangleq \sqrt{\rho_1\rho_2} = \sqrt{U^2 - V^2} \quad . \quad (2.1\text{-}55e)$$

$$\Pr\{\psi_1 < \psi < \psi_2\} = \begin{cases} F(\psi_2) - F(\psi_1) + 1, & \psi_1 < \Delta\phi < \psi_2; \\ F(\psi_2) - F(\psi_1), & \psi_1 > \Delta\phi \text{ or } \psi_2 < \Delta\phi; \end{cases} \quad (2.1\text{-}56a)$$

where

$$F(\psi) = \int_{-\pi/2}^{\pi/2} dx \, \frac{e^{-E(x)}}{4\pi} \left[ \frac{W\sin(\Delta\phi - \psi)}{U - V\sin x - W\cos(\Delta\phi - \psi)\cos x} + \frac{r\sin\psi - \lambda\cos\psi}{1 - (r\cos\psi + \lambda\sin\psi)\cos x} \right] \quad (2.1\text{-}56b)$$

A derivation of (2.1-55) is given in Appendix A which does not use the characteristic function approach.

### 2.1.6 FM Noise Clicks

As the signal plus noise waveform is processed by the digital FM receiver, the total phase derivative is extracted by the FM discriminator. We have seen that this derivative can be expressed by the sum of signal ($\dot{\phi}$) and noise ($\dot{\eta}$) terms. Recall that the phase noise is expressible as

$$\eta(t) = \tan^{-1}\left\{\frac{\nu_2(t)}{\nu_1(t) + \sqrt{2\rho(t)}}\right\} . \tag{2.1-57}$$

Now, when $\sqrt{\rho(t)}$ (related to the instantaneous amplitude of the signal) is small, the probability that the denominator in (2.1.57) can change from a positive value to a negative value becomes significant. The impact of this factor is illustrated in Figure 2.1-6. In part (a) of the figure, $\sqrt{2\rho}$ is relatively large, so that the example random phase trajectory is confined to the right-hand side of the origin. Part (b) shows that, for the same sequence of values of the noises $\nu_1(t)$ and $\nu_2(t)$, when $\sqrt{2\rho}$ is small, an encirclement of the origin occurs.

A complete encirclement of the origin of course results in a rapid $2\pi$ increase in $\eta$, or impulsive value of the derivative $\dot{\eta}(t)$ - heard as a "click" in FM receivers. Once the denominator of (2.1-57) becomes negative and also the numerator changes sign, it is highly likely that a complete encirclement will follow. Therefore, the expected number of encirclements can be computed as the average number of zeros of $\nu_2(t)$, given that $\nu_1(t) + \sqrt{2\rho}$ is negative.

Assuming that a click or encirclement always results when $|\eta|$ exceeds $\pi$, we can estimate the probable number of clicks per unit time as follows. With the help of Figure 2.1-7, we understand that a positive click will occur in the

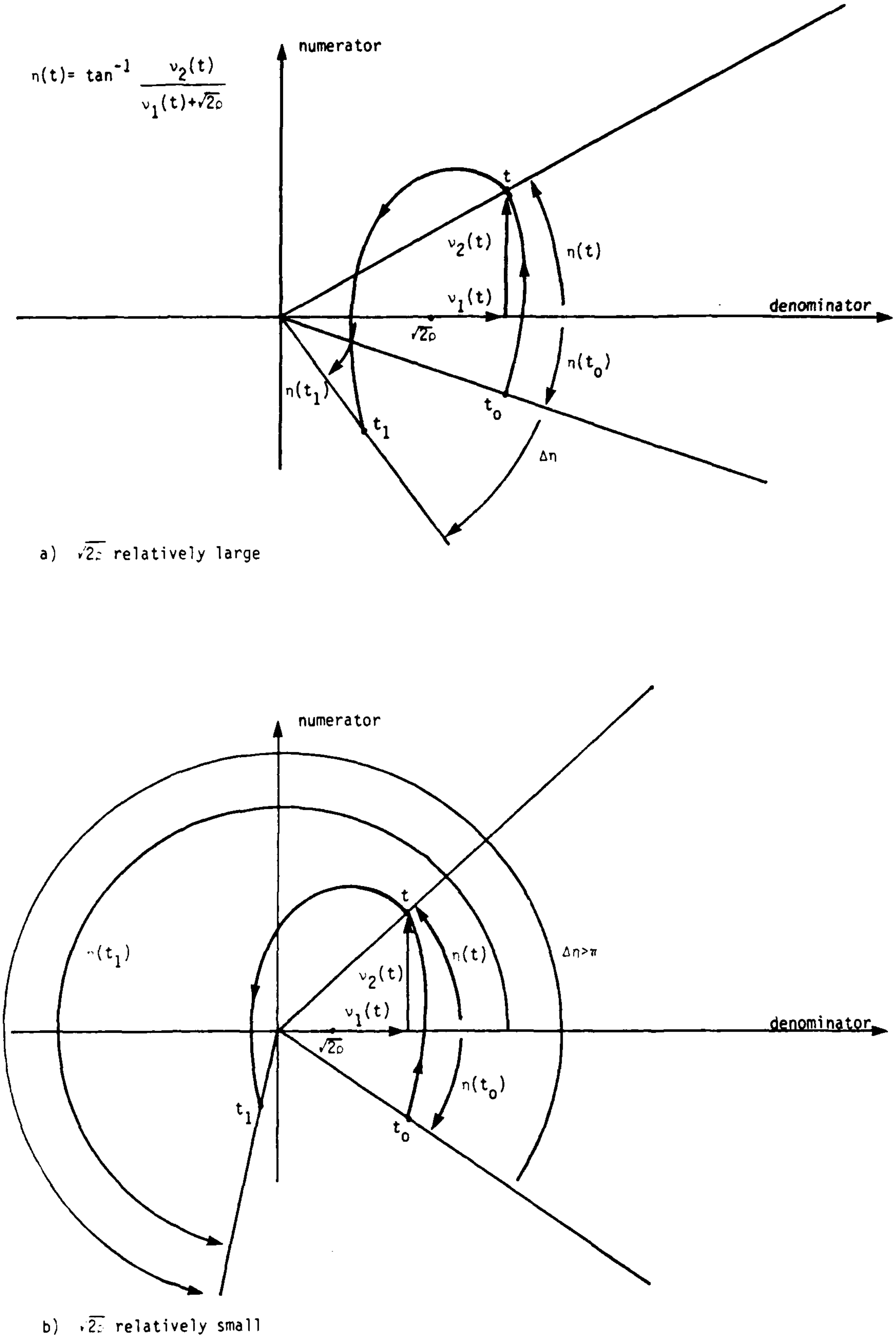
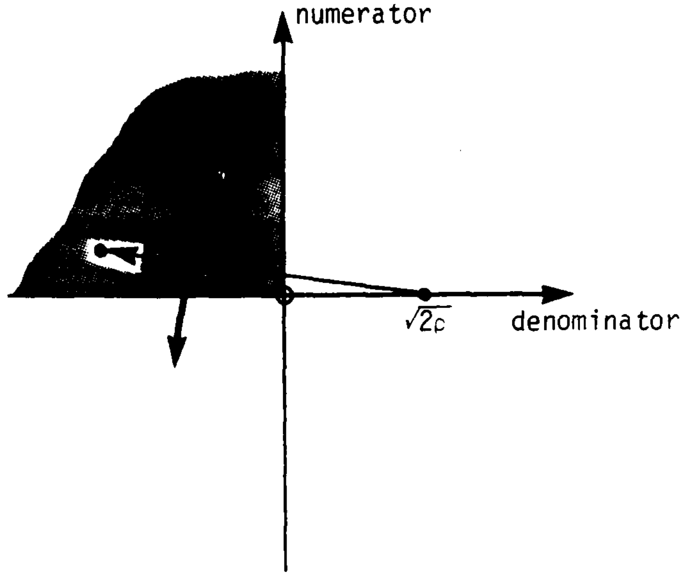


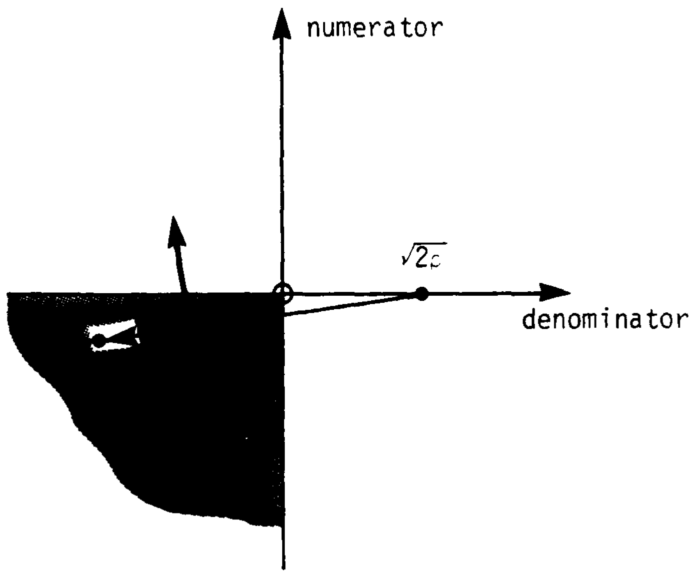

FIGURE 2.1-6 EFFECT OF AMPLITUDE ON LIKELIHOOD OF PHASE ENCIRCLEMENT OF ORIGIN

$$\eta(t) = \tan^{-1}\left\{\frac{v_2}{v_1+\sqrt{2\rho}}\right\}$$




$$\left.\begin{array}{l} 1.\ v_1+\sqrt{2\rho} < 0 \\ 2.\ v_2 > 0 \\ 3.\ v_2+\dot{v}_2 dt < 0 \end{array}\right\} \text{Q II}$$

(a) Conditions for Positive Click




$$\left.\begin{array}{l} 1.\ v_1+\sqrt{2\rho} < 0 \\ 2.\ v_2 < 0 \\ 3.\ v_2+\dot{v}_2 dt > 0 \end{array}\right\} \text{Q III}$$

(b) Conditions for Negative Click

FIGURE 2.1-7 CONDITIONS FOR FM NOISE CLICKS interval (0, dt) if the phase noise angle $\eta$ is in the second quadrant, and if it is increasing fast enough to cross into the third quadrant during the interval. In terms of the quantities in (2.1-57) and their derivatives, the probability of a positive click in (0, dt) can be expressed as

$$P_+ = \Pr\{\nu_1 < -\sqrt{2\rho},\ \nu_2 > 0,\ \nu_2 + \dot{\nu}_2 dt < 0\}$$

$$= \int_{-\infty}^{\sqrt{2\rho}} d\nu_1 \int_{-\infty}^{0} d\dot{\nu}_2 \int_{0}^{-\dot{\nu}_2 dt} d\nu_2\ p(\nu_1,\nu_2,\dot{\nu}_2). \tag{2.1-58}$$

Similarly, a negative click will result when the phase noise goes from the third to the second quadrant, with probability

$$P_- = \Pr\{\nu_1 < -\sqrt{2\rho},\ \nu_2 < 0,\ \nu_2 + \dot{\nu}_2 dt > 0\}$$

$$= \int_{-\infty}^{-\sqrt{2\rho}} d\nu_1 \int_{0}^{\infty} d\dot{\nu}_2 \int_{-\dot{\nu}_2 dt}^{0} d\nu_2\ p(\nu_1,\nu_2,\dot{\nu}_2). \tag{2.1-59}$$

In order to calculate these probabilities, it is necessary to develop the joint pdf in (2.1-58) and (2.1-59).

2.1.6.1 <u>Calculation of Click Rates</u>

Recall that the total normalized waveform is

$$[\sqrt{2\rho(t)} + \nu_1(t)] \cos[\omega_0 t + \phi(t)]$$

$$- \nu_2(t) \sin[\omega_0 t + \phi(t)] \tag{2.1-60a}$$

$$= \sqrt{2\rho(t)} \cos[\omega_0 t + \phi(t)]$$

$$+ \nu_c(t) \cos\omega_0 t - \nu_s(t) \sin\omega_0 t, \tag{2.1-60b}$$

with $(\nu_c, \nu_s)$ being the conventional (normalized) bandpass noise components with

(assuming a symmetrical passband)

$$E\{\nu_c(t)\nu_c(t+\tau)\} = E\{\nu_s(t)\nu_s(t+\tau)\} = r(\tau) \tag{2.1-61a}$$

$$E\{\nu_c(t)\nu_s(t+\tau)\} = -E\{\nu_s(t)\nu_c(t+\tau)\} = 0. \tag{2.1-61b}$$

Then

$$\nu_1(t) = \nu_c(t)\cos\phi(t) + \nu_s(t)\sin\phi(t) \tag{2.1-62a}$$

$$\nu_2(t) = -\nu_c(t)\sin\phi(t) + \nu_s(t)\cos\phi(t), \tag{2.1-62b}$$

and the noise correlation functions are

$$E\{\nu_1(t)\nu_2(t+\tau)\} = r(\tau)\sin[\phi(t)-\phi(t+\tau)]. \tag{2.1-63a}$$

$$E\{\nu_1(t)\nu_1(t+\tau)\} = r(\tau)\cos[\phi(t)-\phi(t+\tau)]$$

$$= E\{\nu_2(t)\nu_2(t+\tau)\}. \tag{2.1-63b}$$

From these expressions we can deduce that the nonzero moments are

$$\sigma_1^2 \triangleq E\{\nu_1^2(t)\} = 1 = E\{\nu_2^2(t)\} \tag{2.1-64a}$$

$$\rho\sigma_1\sigma_2 \triangleq E\{\nu_1(t)\dot{\nu}_2(t)\} = -\dot{\phi}(t) \tag{2.1-64b}$$

$$\sigma_2^2 \triangleq E\{\dot{\nu}_2^2(t)\} = -\frac{\partial^2}{\partial\tau^2} r(\tau)\Big|^{\tau=0} + [\dot{\phi}(t)]^2. \tag{2.1-64c}$$

Since $\nu_1$, $\nu_2$, and $\dot{\nu}_2$ are Gaussian random variables, their joint pdf is then

$$p_1(\nu_1,\dot{\nu}_2,\nu_2) = \frac{1}{\sqrt{2\pi}} e^{-\nu_2^2/2} p_2(\nu_1,\dot{\nu}_2), \tag{2.1-65a}$$

where

$$p_2(\nu_1,\dot{\nu}_2) = \frac{1}{2\pi\sigma_2\sqrt{1-\xi^2}} \exp\left\{ -\frac{1}{2(1-\xi^2)}\left[\nu_1{}^2 - 2\xi\left(\frac{\nu_1\dot{\nu}_2}{\sigma_2}\right) + \left(\frac{\dot{\nu}_2}{\sigma_2}\right)^2\right]\right\}. \quad (2.1\text{-}65b)$$

With this joint pdf we can now calculate the click probabilities $P_+$ and $P_-$.

First we note that

$$\int_0^{-\dot{\nu}_2 dt} d\nu_2\, p_1(\nu_1,\nu_2,\dot{\nu}_2) = \left[\frac{1}{2} - Q(-\dot{\nu}_2 dt)\right] p_2(\nu_1,\dot{\nu}_2) \quad (2.1\text{-}66a)$$

$$= \frac{1}{2}\,\mathrm{erf}\left(\frac{-\dot{\nu}_2 dt}{\sqrt{2}}\right) p_2(\nu_1,\dot{\nu}_2) \quad (2.1\text{-}66b)$$

and

$$\int_{-\dot{\nu}_2 dt}^{0} d\nu_2\, p_1(\nu_1,\nu_2,\dot{\nu}_2) = \left[\frac{1}{2} - Q(\dot{\nu}_2 dt)\right] p_2(\nu_1,\dot{\nu}_2) \quad (2.1\text{-}66c)$$

$$= \frac{1}{2}\,\mathrm{erf}\left(\frac{\dot{\nu}_2 dt}{\sqrt{2}}\right) p_2(\nu_1,\dot{\nu}_2)\ . \quad (2.1\text{-}66d)$$

Anticipating that we will make the value of dt very small, a suitable approximation is the first order MacLaurin series

$$Q(x) \simeq \frac{1}{2} - x \cdot \frac{e^{-x^2/2}}{\sqrt{2\pi}}\ . \quad (2.1\text{-}67)$$

Substitution of this approximation in (2.1-66) and then in (2.1-58) and (2.1-59) gives

$$P_+ \simeq \int_{-\infty}^{-\sqrt{2}\ldots} d\nu_1 \int_{-\infty}^{0} d\dot{\nu}_2\, (-\dot{\nu}_2 dt)\, \frac{e^{-(\dot{\nu}_2 dt)^2/2}}{\sqrt{2\pi}}\, p_2(\nu_1,\dot{\nu}_2) \quad (2.1\text{-}68a)$$

and

$$P_- \simeq \int_{-\infty}^{-\sqrt{2}\ldots} d\nu_1 \int_{0}^{\infty} d\dot{\nu}_2\, (\dot{\nu}_2 dt)\, \frac{e^{-(\dot{\nu}_2 dt)^2/2}}{\sqrt{2\pi}}\, p_2(\nu_1,\dot{\nu}_2). \quad (2.1\text{-}68b)$$

The net positive click rate then can be written as

$$\dot{N}_c \triangleq \lim_{dt \to 0} \frac{P_+ - P_-}{dt} \tag{2.1-69a}$$

$$= -\int_{-\infty}^{-\sqrt{2\rho}} d\nu_1 \int_{-\infty}^{\infty} d\dot{\nu}_2 \, \dot{\nu}_2 \, p_2(\nu_1, \dot{\nu}_2) \frac{1}{\sqrt{2\pi}}$$

or

$$\dot{N}_c = \frac{\sigma_{\dot{\nu}}\dot{\xi}}{2\pi} e^{-\rho} = -\frac{\dot{\phi}}{2\pi} e^{-\rho}. \tag{2.1-69b}$$

From this expression we observe that the tendency is for clicks to oppose the direction of rotation (phase) of the signal. Averaging over the T-second data bit interval results in the following expected number of positive clicks:

$$\overline{N}_c = \frac{-1}{2\pi} \int_{t_s - T}^{t_s} dt \, \dot{\phi}(t) e^{-\rho(t)}. \tag{2.1-70}$$

2.1.6.2 Effect of Clicks on the Phase Distribution

The differential phase $\Delta\phi$ at the output of the digital FM receiver was shown earlier to have the probability density function $p_{\Delta\phi}(x)$. Now we must say that this previous result pertains to the case of no clicks, or to the portion of the differential phase excluding clicks. If a discrete distribution for the number of clicks in $(t_s - T, t_s)$ is postulated, then the pdf for the differential phase including clicks can be written as

$$p_d(x) \equiv p_d(x; \Delta\phi) = \sum_{n=0}^{\infty} \Pr\{N = n\} \, p_{\Delta\phi}[x + 2\pi n \, \mathrm{sgn}(\Delta\phi)]. \tag{2.1-71}$$

In this expression we use the fact that the sign of $\Delta\phi$ is the same as the sign of $\dot{\phi}(t)$ in the bit interval, and assume that only clicks opposite in sign to $\Delta\phi$

are significant. In (2.1-71) also, $N$, the net number of opposing clicks, is assumed to have the Poisson distribution:

$$\Pr\{N = n\} = \exp\{-|\overline{N}_c|\} \cdot |\overline{N}_c|^n/n! \ . \qquad (2.1\text{-}72)$$

Figure 2.1-8 illustrates the effect of opposing clicks on the differential phase probability distribution.

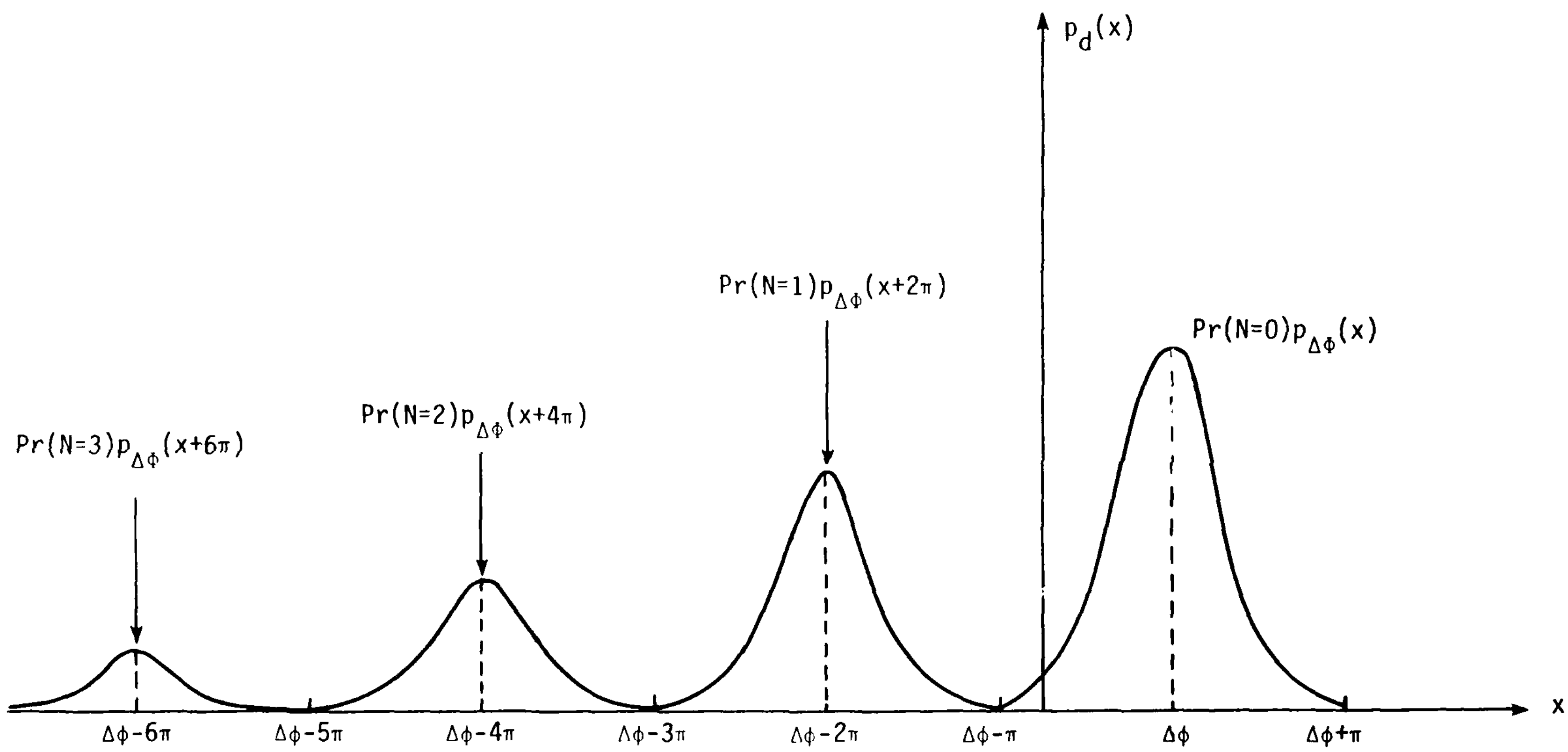



FIGURE 2.1-8 PROBABILITY DENSITY FUNCTION FOR DIFFERENTIAL PHASE INCLUDING FM CLICKS

## 2.2 ERROR PROBABILITY FORMULATION

In this subsection, we discuss the bit decision procedures assumed to be implemented by the FH/CPFSK receiver, and derive the basic expressions needed to evaluate uncoded bit error probability (BER) in consideration of thermal noise, clicks, jamming noise, and intersymbol interference.

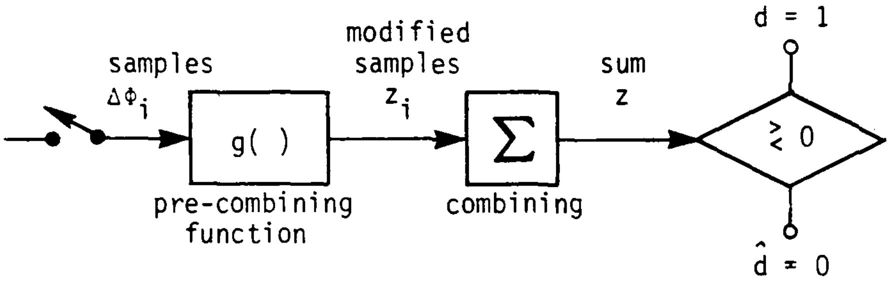
### 2.2.1 Decision Processing

In Section 2.1 we have discussed how the limiter-discriminator receiver with integrate-and-dump filtering develops samples of the differential phase. We denote those samples by $\{\Delta\phi_i, i=1,2,\ldots,L\}$, where L is number of hops on which the data is repeated for possible diversity improvement against the jamming. If $L>1$, the T-second period over which each sample is developed is related to the bit period by $T = T_b/L$. In concept, we can diagram the decision processing to be evaluated in this report as shown in Figure 2.2-1(a): the samples are first processed by a pre-combining function $g(\cdot)$ to produce modified samples $\{z_i\}$, where

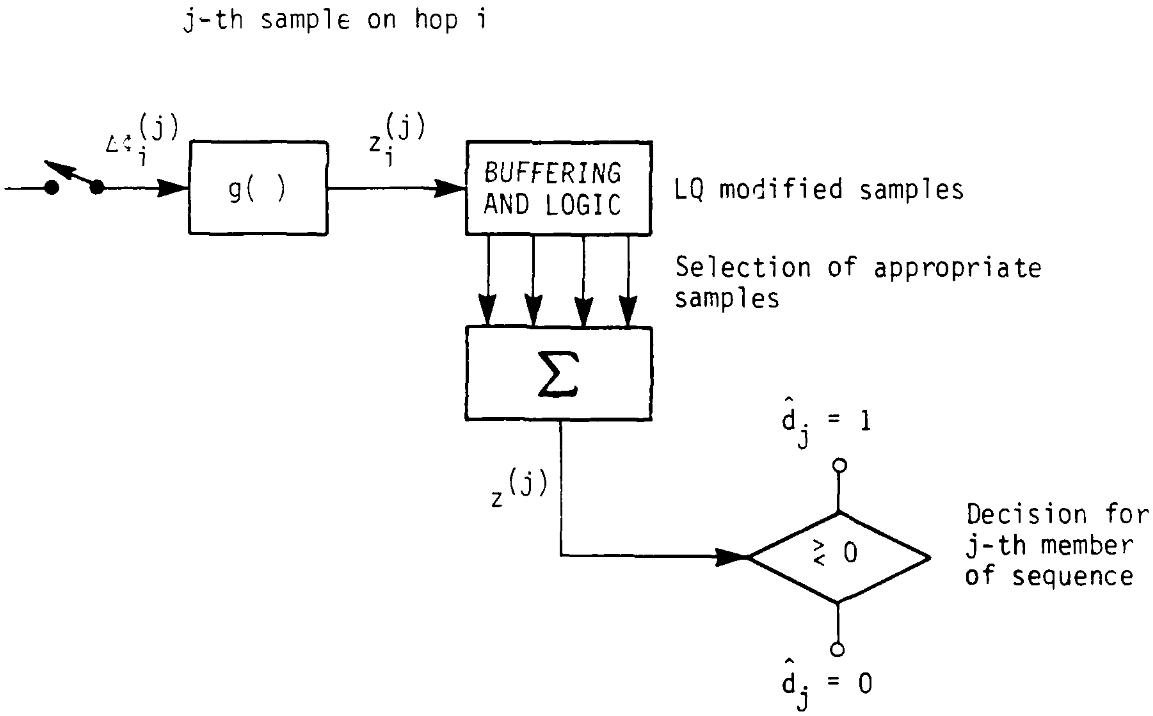
$$z_i = g(\Delta\phi_i). \tag{2.2-1}$$

For example, $g(\cdot)$ may represent analog-to-digital conversion with a specific number of quantization levels; if the number of levels is just two, then the function $g(\cdot)$ implements the "hard decision" given by

$$z_{iHD} \triangleq \begin{cases} +1 & , \Delta\phi_i \geq 0; \\ -1 & , \Delta\phi_i < 0. \end{cases} \tag{2.2-2}$$




(a) Conceptual Form of Decision Processing




(b) Details for Slow-hopping System

FIGURE 2.2-1 RECEIVER DECISION PROCESSING

The modified samples are then summed to produce a single decision statistic z, that is,

$$z \triangleq \sum_{i=1}^{L} z_i = \sum_{i=1}^{L} g(\Delta\Phi_i), \tag{2.2-3}$$

so that the final bit decision $\hat{d}$ is made by examining the sign of z:

$$\hat{d} = \begin{cases} 1, & z \geq 0 \\ 0, & z \leq 0. \end{cases} \tag{2.2-4}$$

In practice, since the data rate is assumed to be higher than the hopping rate, the L "chips" constituting the repetitions of each "data bit" are not transmitted one after another, as implied by Figure 2.2-1(a). Instead, they are transmitted as members of sequences of Q chips per hop, and the sequences are repeated over L different hops with the idea that some of the hops will evade the jamming. Thus, some prearranged formatting is necessary so that the receiver can retrieve the L "pieces" of a given bit. Figure 2.2-1(b) illustrates the processing necessary to collect a particular data bit's L "pieces" from among the set of LQ modified samples which are received over the L hops. For our analyses, it is not necessary to describe the receiver decision processing any further, except to note that scrambling or re-ordering of the Q-bit sequence from hop to hop is an option; if such a re-ordering is done, then the intersymbol interference differs from hop to hop, and this affects the analysis, as we shall mention below at the appropriate place.

### 2.2.2 Conditional Error Probabilities

Each of a data bit's L differential phase samples $\Delta\Phi_i$ is a random variable whose distribution is parametric in the values of the following quantities during the sample interval: the CNR $\rho^{(i)}$, representing the thermal noise and the presence or absence of jamming; and the nominal differential phase $\Delta\phi_i$ and the CNR-related parameters $U_i$, $V_i$, and $W_i$, representing the data pattern-dependent intersymbol interference effects. Thus the distribution of the modified sample $z_i$ is conditioned on these quantities as well:

$$p_{z_i}(\alpha|\rho^{(i)}, \underline{\xi}_i) \equiv p_{z_i}(\alpha|\rho^{(i)}, U_i, V_i, W_i, \Delta\phi_i), \tag{2.2-5}$$

where the vector $\underline{\xi}_i$ is a shorthand notation for the set of data-dependent parameters listed.

The symmetries of the data-dependent parameters are such that

$$\Pr\{\Delta\Phi_i < 0|\rho^{(i)}, x1y\} = \Pr\{\Delta\Phi_i > 0|\rho^{(i)}, \bar{x}\,0\bar{y}\}, \tag{2.2-6}$$

where "x1y" stands for a data pattern with a data value of 1 at the sample time, and "$\bar{x}0\bar{y}$" is the complementary pattern. If the pre-combining processing function $g(\cdot)$ is perfectly symmetric (odd), then it is also true that

$$\Pr\{z_i < 0|x1y\} = \Pr\{z_i > 0|\bar{x}0\bar{y}\}, \tag{2.2-7}$$

and for L=1 we may restrict our attention to the x1y patterns in calculating the BER.

For $L>1$, the randomness of the $\{z_i\}$ is due to noise power at different portions of the RF spectrum, so they are statistically independent, with their joint pdf given by

$$\prod_{i=1}^{L} p_{z_i}(\alpha_i|\rho^{(i)}, \underline{\beta}_i) \,. \tag{2.2-8}$$

The parameter sets $\{\rho^{(i)}, \underline{\beta}_i\}$ are in general different for each hop, with the L values of $\rho^{(i)}$ independent of each other since it is assumed that the event of jamming on one hop is independent of the event of jamming on another hop. The data dependent parameter values, however, are related, since the data sample values are the same. There are two cases we shall consider: (1) the transmitted data sequences on the L hops are identical, and (2) the transmitted data sequences are scrambled. For case (1), the data-dependent parameters are identical ($\underline{\beta}_i = \underline{\beta}$ , all i). For case (2), the individual $\underline{\beta}_i$ are independently selected from those corresponding to the four possible data patterns represented by x1y, or x0y.

In consideration of the symmetries we have discussed with respect to the distribution of individual hop samples, it is possible to state the symmetry which applies to the conditional probability of error as follows. The distribution of the sum decision statistic is such that when $\lambda = 0$

$$\begin{aligned} &\Pr\{z > 0|\rho^{(1)}, \underline{\beta}_1;\rho^{(2)}, \underline{\beta}_2;\ldots;\rho^{(L)} \underline{\beta}_L\}, \\ &= \Pr\{z < 0|\rho^{(1)}, \overline{\underline{\beta}}_1;\rho^{(2)}, \overline{\underline{\beta}}_2;\ldots;\rho^{(L)}, \overline{\underline{\beta}}_L\}, \end{aligned} \tag{2.2-9}$$

where $\overline{\underline{\beta}}_i$ denotes the data-dependent parameter values arising from the complement of the pattern which produces $\underline{\beta}_i$. Therefore, the conditional probabilities of error have the symmetric relationship

$$P(e|\rho^{(1)}, x_1 1 y_1;\ldots;\rho^{(L)}, x_L 1 y_L)$$

$$= P(e|\rho^{(1)}, \bar{x}_1 0 \bar{y}_1;\ldots;\rho^{(L)}, \bar{x}_L 0 \bar{y}_L), \qquad (2.2\text{-}10)$$

when $\lambda = 0$, which holds for a noise spectrum symmetric about the carrier frequency.

### 2.2.3 Unconditional Error Probability

The unconditional error probability is found by averaging the conditional error probability with respect to the jamming events and the intersymbol interference patterns. The required pdf's for averaging are assumed to be discrete-valued.

The averaging over the jamming events is most easily done at the per-hop level. That is, in the calculation of the error, we use the hop pdf's

$$p_{z_i}(\alpha|\underline{\beta}_i) = \sum_R \Pr\{\rho^{(i)} = R\}\, p_{z_i}(\alpha|\rho^{(i)}, \underline{\beta}_i) \qquad (2.2\text{-}11a)$$

$$= (1-\gamma)\, p_{z_i}(\alpha|\rho_N, \underline{\beta}_i) + \gamma p_{z_i}(\alpha|\rho_T, \underline{\beta}_i), \qquad (2.2\text{-}11b)$$

where the unjammed SNR $\rho_N$ and the jammed SNR $\rho_T$ are related to the signal bit energy and noise spectral densities by

$$\rho_N = \frac{1}{L} \cdot \frac{E_b}{N_0} \qquad (2.2\text{-}12a)$$

$$\rho_T = \frac{1}{L} \cdot \frac{E_b}{N_0+N_{0J}} = \frac{1}{L} \cdot \frac{E_b}{N_0+N_J/\gamma} \qquad (2.2\text{-}12b)$$

In (2.2-12b), it is assumed that the actual jamming spectral density $N_{0J}$ is related to the average jamming spectral density $N_J$ by $N_{0J} = N_J/\gamma$. This is turn assumes that the fraction $\gamma$ of the hopping band is jammed with noise density $N_{0J}$; the fraction $\gamma$ also is the probability of jamming, as used in (2.2-11b).

Assuming that the partially-averaged pdf's given in (2.2-11) are used, the conditional probabilities of concern are of the form

$$P(e|x_1 1 y_1, \ldots, x_L 1 y_L). \tag{2.2-13}$$

The complementary patterns need not be considered in view of the symmetries discussed above, assuming that 1's and 0's are equiprobable. Therefore, the general BER is the average

$$P(e) = \sum_{\underline{x},\underline{y}} \left(\frac{1}{4}\right)^L P(e|x_1 1 y_1, \ldots, x_L 1 y_L), \tag{2.2-14}$$

if the data sequences are scrambled from hop to hop. If identical data sequences are used, then $x_i = x$ and $y_i = y$, and

$$P(e) = \frac{1}{4}\{P(e|010) + P(e|011) + P(e|110) + P(e|111)\}. \tag{2.2-15}$$

As a practical matter, if the sequences are scrambled, it is more convenient to use the marginal (averaged) sample pdf's

$$p_z(\alpha) = \frac{1}{4}\{p_{z_i}[\alpha|\underline{\beta}(010)] + p_{z_i}[\alpha|\underline{\beta}(011)] + p_{z_i}[\alpha|\underline{\beta}(110)] + p_{z_i}[\alpha|\underline{\beta}(111)]\}, \tag{2.2-16}$$

and to straightforwardly calculate

$$P(e) = Pr\{\sum z_i < 0|1\}. \tag{2.2-17}$$

### 2.2.4 Treatment of Clicks

The single-hop pdf given by (2.2-11) may be expanded in terms of the number of FM noise clicks, as follows:

$$
\begin{aligned}
p_{z_i}(\alpha|\underline{\beta}) &= (1-\gamma)p_{z_i}(\alpha|\rho_N,\underline{\beta}) + \gamma p_{z_i}(\alpha|\rho_T,\underline{\beta}) \\
&= (1-\gamma)\sum_{n=0}^{\infty} \Pr\{N_c=n|\rho_N,\underline{\beta}\}\ f(\alpha+2\pi n;\rho_N,\underline{\beta}) \\
&\quad + \gamma \sum_{n=0}^{\infty} \Pr\{N_c=n|\rho_T,\underline{\beta}\}\ f(\alpha+2\pi n;\rho_T,\underline{\beta}),
\end{aligned}
\qquad (2.2\text{-}18)
$$

with $N_c$ the number of clicks. For convenience we consider only patterns with $\Delta\omega > 0$, so that the probable clicks of significance are negative, those which have the effect of shifting the differential phase pdf to the left by some multiple of $2\pi$. *In (2.2-18) we indicate an infinite sum with respect to the* number of clicks in the interval T; this can be interpreted as averaging the pdf $f(\alpha+2\pi N;\rho,\underline{\beta})$ with respect to N. We also, in using $\underline{\beta}$ (without a subscript), suppose that *no* scrambling of the data is done from hop to hop; then (2.2-18) is the pdf for each of the L sampled differential phases, conditioned on the data pattern.

#### 2.2.4.1 Number of Significant Clicks

It is possible to write (2.2-18) as

$$
p_{z_i}(\alpha|\underline{\beta}) = \sum_{n=0}^{\infty} f_n^{(1)}\ (\alpha+2\pi n;\underline{\beta}), \qquad (2.2\text{-}19)
$$

emphasizing the fact that this pdf is the superposition of $2\pi$ translations of pdf's for given numbers of clicks. We then can appreciate that the sum z also has a pdf which is of this form, that is, is a superposition of (overlapping) pdf's for given numbers of clicks. When we consider that each of these constituent pdf's is nonzero for only a finite portion of the z-axis, then it follows that a certain, finite number are nonzero for positive z. This fact allows us to restrict our attention to a finite number of clicks when computing the probability of error.

The reasoning is as follows: the sum's pdf may be expressed by

$$p_z(\alpha|\underline{\beta}) = \sum_{n=0}^{\infty} f_n^{(L)}(\alpha+2\pi n;\underline{\beta})$$

$$= \sum_{n=0}^{\infty} g_n[\alpha-L\Delta\phi+2\pi n;\underline{\beta}], \qquad (2.2\text{-}20)$$

in which the $g_n(\alpha)$ are pdf's centered at $\alpha = 0$, so that the constituent pdf's $f_n^{(L)}(\cdot)$ are seen to be centered at $L\Delta\phi-2\pi n$.

If the single-hop pdf's are nonzero for a $2\pi$ interval, then it follows that the $g_n$ are nonzero for $2\pi L$ intervals as a result of convolution of L pdf's. Now, the error probability can be computed from

$$P(e) = \Pr\{z<0\}$$

$$= 1 - \Pr\{z>0\}. \qquad (2.2\text{-}21)$$

Therefore, in computing P(e) we need only those constituent pdf's which are nonzero for $z > 0$, corresponding to click numbers $n = 0$ to $n = n_{max}$, where nmax is deduced from

$$L\Delta\phi - 2\pi\, \text{nmax} + L\pi < 0$$

or

$$\text{nmax} > \frac{L}{2\pi}(\Delta\phi + \pi). \tag{2.2-22a}$$

For example if $\Delta\phi = \pi/2$, then nmax is the smallest integer larger than 3L/4. It is further deduced that

$$\begin{aligned} \text{nmax} &= \left\lfloor \frac{L}{2}\left(1 + \frac{\Delta\phi}{\pi}\right)\right\rfloor \\ &= \begin{cases} L-1, & \Delta\phi/\pi > \dfrac{L-2}{L}, \\ L-2, & \Delta\phi/\pi < \dfrac{L-2}{L}, \end{cases} \end{aligned} \tag{2.2-22b}$$

for $0 < \Delta\phi < \pi$.

Figure 2.2-2 illustrates for $\Delta\phi \simeq .6\pi$ the facts that for L=1, the P(e) can be computed using just the pdf for no clicks; for L=2, using the sum pdf's for zero and one click; and for L=3, using zero, one, and two clicks.

Knowing this basic rule on the number of clicks that are significant for computing the error saves much computation, and also permits simplification of the analysis somewhat.

### 2.2.4.2 Application to Sum Characteristic Function

With the preceding in mind we can truncate the click-indexed series (2.2-19) and write the single-hop pdf as

$$p_{z_i}(\alpha|\underline{\beta}) = \sum_{n=0}^{\text{nmax}} f_n^{(1)}(\alpha + 2\pi n; \underline{\beta}), \tag{2.2-23}$$

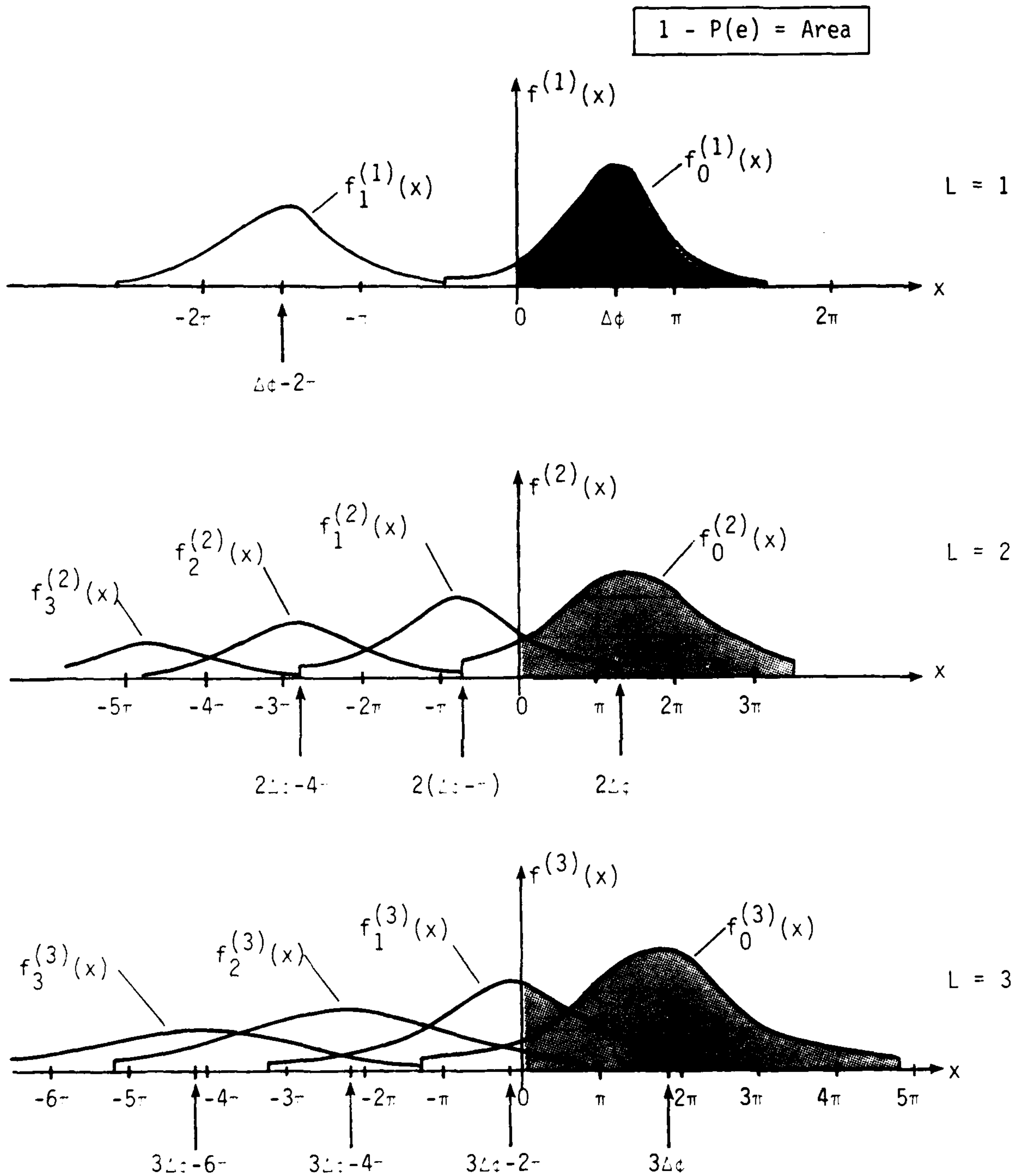



FIGURE 2.2-2 ILLUSTRATION OF SUM PDF CONSTITUENT FUNCTIONS AND THOSE WHICH ARE SIGNIFICANT FOR COMPUTING THE ERROR

with characteristic function

$$\varphi^{(1)}(\nu) = \sum_{n=0}^{nmax} \varphi_n(\nu) e^{-j2\pi n\nu}, \qquad (2.2\text{-}24a)$$

where

$$\varphi_n(\nu) \triangleq \mathcal{F}\left\{f_n^{(1)}(\alpha;\underline{\beta})\right\}. \qquad (2.2\text{-}24b)$$

Then, since the samples are identically distributed (given $\underline{\beta}$), the characteristic function for the sum of L samples is

$$\begin{aligned}\varphi^{(L)}(\nu) &= [\varphi^{(1)}(\nu)]^L \\ &= \left[\sum_{n=0}^{nmax} \varphi_n(\nu) e^{-j2\pi n\nu}\right]^L \\ &= \sum_{\underline{\ell}:\Sigma\ell_n = L} \binom{L}{\ell_0, \ell_1, \ldots, \ell_{nmax}} \prod_{k=0}^{nmax} \varphi_k^{\ell k} e^{-j2\pi\nu k\ell_k}. \qquad (2.2\text{-}25)\end{aligned}$$

The first several terms of (2.2-25) are

$$\begin{aligned}\varphi^{(L)}(\nu) &= \varphi_0^L + L\varphi_0^{L-1} \varphi_1 e^{-j2\pi\nu} \\ &+ \left[\binom{L}{2}\varphi_0^{L-2} \varphi_1^2 + L\varphi_0^{L-1} \varphi_2\right] e^{-j4\pi\nu} \qquad (2.2\text{-}26) \\ &+ \ldots.\end{aligned}$$

## 3.0 BER CALCULATIONS FOR DIVERSITY SUM

In this section, we develop theoretical expressions and numerical results for the BER produced by the limiter-discriminator FH/CPFSK receiver using straightforward linear combining of the L diversity samples of the differential phase. Worst-case partial-band noise jamming (WCPBNJ) is assumed, and our interest is to determine the improvement in the BER, if any, that is achieved by processing the diversity samples in this manner.

### 3.1 L = 1 BER IN WORST-CASE PARTIAL-BAND NOISE JAMMING

The baseline case for measuring improvement in performance is the case of no diversity, that is, L = 1. The BER for this case is

$$\left\langle \Pr\{z = \Delta\phi < 0 \mid x1y\} \right\rangle_{x,y} \qquad (3.1\text{-}1)$$

where as before "x1y" denotes the different data patterns which determine the intersymbol interference parameters, and the effects of jamming and noise clicks are already included. The pdf, conditioned on the pattern, is given by

$$p_z(\alpha,\underline{\beta}) = (1-\gamma)\sum_n p_{0,n}\, f(\alpha+2\pi n;\rho_N,\underline{\beta}) + \gamma \sum_n p_{1,n}\, f(\alpha+2\pi n;\rho_T,\underline{\beta}) \qquad (3.1\text{-}2a)$$

$$= \sum_n f_n^{(1)}(\alpha+2\pi n;\underline{\beta}) \triangleq p^{(1)}(\alpha|\underline{\beta}), \qquad (3.1\text{-}2b)$$

in the notation of Section 2.2.4, with the click probabilities $p_{0,n}$ (unjammed) and $p_{1,n}$ (jammed) given by

$$p_{i,n} \equiv p_{i,n}(\underline{\beta}) = \Pr\{N_c = n \mid \rho_i, \underline{\beta})\}. \tag{3.1-3}$$

The conditional BER is conveniently expressed as

$$\begin{aligned} P(e \mid x1y) &= 1 - \Pr\{z > 0 \mid x1y\} \\ &= 1 - \int_0^{\Delta\phi+\pi} d\alpha\, p_z(\alpha \mid \underline{\beta}) \\ &= 1 - \int_0^{\Delta\phi+\pi} d\alpha\, f_0^{(1)}(\alpha; \underline{\beta}). \end{aligned} \tag{3.1-4}$$

Note that only the term of the pdf corresponding to zero clicks is needed. Expanding further yields

$$\begin{aligned} P(e\, x1y) &= 1 - (1-\gamma)p_{0,0} \int_0^{\Delta\phi+\pi} d\alpha\, f(\alpha; \rho_N, \underline{\beta}) \\ &\quad - \gamma\, p_{1,0} \int_0^{\Delta\phi+\pi} d\alpha\, f(\alpha; \rho_T, \underline{\beta}) \\ &= 1 - (1-\gamma)\, p_{0,0} - \gamma\, p_{1,0} \\ &\quad - (1-\gamma)p_{0,0}\, [F(\Delta\phi+\pi; \rho_N, \underline{\beta}) - F(0; \rho_N, \underline{\beta})] \\ &\quad - \gamma\, p_{1,0}\, [F(\Delta\phi+\pi; \rho_T, \underline{\beta}) - F(0; \rho_T, \underline{\beta})], \end{aligned} \tag{3.1-5}$$

where the function $F(\cdot)$ is given in (2.1-5b), with $\lambda = 0$.

### 3.1.1 Case of No Jamming

With no jamming, the BER is (3.1-5) with the substitution of $\gamma = 0$, resulting in

$$P(e|x1y) = 1 - p_{0,0} - p_{0,0}\, F(\Delta\phi+\pi;\rho_N,\underline{\beta}) + p_{0,0}\, F(0;\rho_N,\underline{\beta}). \qquad (3.1\text{-}6)$$

Averaging over the data patterns (with parameters denoted by $\underline{\beta}$) gives

$$P(e) = \frac{1}{4}\{P(e|010) + P(e|011) + P(e|110) + P(e|111)\}. \qquad (3.1\text{-}7)$$

#### 3.1.1.1 Pattern-dependent Quantities

To illustrate the use of the various parameters developed in Section 2.1, we list in full the pattern-dependent quantities:

010 pattern

$$p_{0,0} = e^{-\rho \overline{N}_c}, \qquad (3.1\text{-}8a)$$

where

$$\overline{N}_c = \frac{-1}{2\pi}\int_{-T}^{0} dt\, \frac{\dot{u}(t)v(t) - u(t)\dot{v}(t)}{u^2(t) + v^2(t)}\, \exp\{-\rho[u^2(t) + v^2(t)]\} \qquad (3.1\text{-}8b)$$

and $u(t) = c_1 \cos(\pi t/T)$ (3.1-8c)

$$v(t) = c_2 - c_3 \cos(2\pi t/T). \qquad (3.1\text{-}8d)$$

For computation it is convenient to define the following functions:

$$u'(t) \triangleq u(Tt), \; v'(t) = v(Tt) \tag{3.1-9a}$$

$$w(t) \triangleq T\dot{u}(Tt) \;, \; z(t) \triangleq T\dot{v}(Tt). \tag{3.1-9b}$$

With these functions, (3.1-8b) becomes

$$\overline{N}_c = \frac{-1}{2\pi} \int_{-1}^{0} dx \, \frac{w(x)v'(x) - u'(x)z(x)}{(u'(x))^2 + (v'(x))^2} \, \exp\{-\rho[(u'(x))^2 + (v'(x))^2]\} \,. \tag{3.1-10}$$

The functions u', v', w, and z for the several patterns are given in Table 3.1-1.

Now, the parameter set $\underline{\xi}$ includes the nominal differential phase $\Delta\phi$, and the CNR-related parameters U, V, and W defined in Table 2.1-3. For the 010 pattern, these are

$$\begin{aligned} \Delta\phi(010) &= 2 \tan^{-1}\left(\frac{c_1}{c_2 - c_3}\right) = 1.2239 \text{ radians} \\ U(010) &= \tfrac{1}{2}[c_1^2 + (c_2 - c_3)^2] = 0.86965 \\ V(010) &= 0 \\ W(010) &= U(010). \end{aligned} \tag{3.1-11}$$

The numerical values assume that h = 0.7, $W_{IF}T = 1.0$, and a Gaussian shaped I.F. filter.

TABLE 3.1-1 PATTERN-DEPENDENT FUNCTIONS FOR COMPUTING THE ERROR PROBABILITY

| PATTERN | $u'(x)$ | $v'(x)$ |
|---|---|---|
| 010 | $c_1 \cos(\pi x)$ | $c_2 - c_3 \cos(2\pi x)$ |
| 011 | $c_4 \sin(\pi x/2) - c_5 \sin(3\pi x/2)$ | $c_6 + c_7 \cos(\pi x) - c_8 \cos(2\pi x)$ |
| 110 | $c_4 \cos(\pi x/2) + c_5 \cos(3\pi x/2)$ | $c_6 - c_7 \cos(\pi x) - c_8 \cos(2\pi x)$ |
| 111 | $a_0 \sin(\pi hx)$ | $a_0 \cos(\pi hx)$ |
| | $w(x)$ | $z(x)$ |
| 010 | $- c_1 \pi \sin(\pi x)$ | $2\pi c_3 \sin(2\pi x)$ |
| 011 | $\frac{c_4 \pi}{2} \cos(\pi x/2) - \frac{3c_5 \pi}{2} \cos(3\pi x/2)$ | $-c_7 \pi \sin(\pi x) + 2\pi c_8 \sin(2\pi x)$ |
| 110 | $- \frac{c_4 \pi}{2} \sin(\pi x/2) - \frac{3c_5 \pi}{2} \sin(3\pi x/2)$ | $c_7 \pi \sin(\pi x) + 2\pi c_8 \sin(2\pi x)$ |
| 111 | $a_0 \pi h \cos(\pi hx)$ | $-a_0 \pi h \sin(\pi hx)$ |

Note: Values of the coefficients are given in Tables 2.1-1 and 2.1-2.

To calculate $P(e|010)$, $p_{0,0}$ is computed by numerical integration using (3.1-8a) and (3.1-10); the $\beta$ parameters are calculated using (3.1-11) and Table 2.1-1, and substituted into numerical integrations of (2.1-56).

011 Pattern

For this pattern (3.1-8a) and (3.1-8b) apply, with u(t) and v(t) given by

$$u(t) = c_4 \sin(\pi t/2T) - c_5 \sin(3\pi t/T) \tag{3.1-12a}$$

$$v(t) = c_6 + c_7 \cos(\pi t/T) - c_8 \cos(2\pi t/T). \tag{3.1-12b}$$

The transformed versions of these functions and their derivatives, shown in Table 3.1-1, are used in (3.1-10) to compute the click number average $\bar{N}_c$.

The parameters needed for computing $F(\cdot)$ are, from Table 2.1-3,

$$\Delta\phi(011) = \tan^{-1}\left[\frac{c_4+c_5}{c_6-c_7-c_8}\right] = 1.7108 \text{ radians}$$

$$U(011) = [(c_4+c_5)^2/2 + c_7^2 + (c_6-c_8)^2] = 0.7784$$

$$V(011) = [2c_7(c_6-c_8) - (c_4+c_5)^2/2] = -\,0.0997 \tag{3.1-13}$$

$$W(011) = \sqrt{U^2(011) - V^2(011)} = 0.7719.$$

110 Pattern

The functions $u(t)$ and $v(t)$ for this pattern are

$$u(t) = c_4 \cos(\pi t/2T) + c_5 \cos(3\pi t/2T) \tag{3.1-14a}$$

and

$$v(t) = c_6 - c_7 \cos(\pi t/T) - c_8 \cos(2\pi t/T), \tag{3.1-14b}$$

with the transformed versions shown in Table 3.1-1. We note that since, for example,

$$u(t;110) = u(t+T;011) = u(-t-T;011) \tag{3.1-15a}$$

$$v(t;110) = v(t+T;011) = v(-t-T;011), \tag{3.1-15b}$$

there is a great deal of symmetry with the pattern 011. In fact, using (3.1-15) we see that

$$\begin{aligned}\overline{N}_c(110) &= \frac{-1}{2\pi}\int_{-T}^{0} dt\ \dot{a}(-t-T;011)e^{-\beta a^2(-t-T;011)} \\ &= \frac{-1}{2\pi}\int_{0}^{T} dt\ \dot{a}(t-T;011)e^{-\beta a^2(t-T;011)} \\ &= \frac{-1}{2\pi}\int_{-T}^{0} dt\ \dot{a}(t;011)e^{-\beta a^2(t;011)} = \overline{N}_c(011).\end{aligned} \tag{3.1-16}$$

We also note from Table 2.1-3 that $\Delta\omega$, U, and W are the same for pattern 110 as for pattern 011. The sign of V changes, but this is of no significance when one examines equation (2.1-56) under a change of the variable of integration from x to -x. Therefore, we can conclude that

$$P(e|110) = P(e|011), \tag{3.1-17}$$

and avoid the necessity for computing $P(e|110)$ separately.

<u>111</u> <u>Pattern</u>

This "all 1's" pattern has the simple result that

$$\overline{N}_c = -\frac{h}{2} e^{-\rho a_0^2}, \tag{3.1-18}$$

since $u(t) = a_0 \sin(\pi ht/T)$ and $v(t) = a_0 \cos(\pi ht/T)$. The pattern parameters for calculating $F(\cdot)$ are

$$U(111) = W(111) = a_0^2\rho = 0.6806\rho$$

$$V(111) = 0 \tag{3.1-19}$$

$$\Delta\phi(111) = \pi h = 2.1991 \text{ radians.}$$

In view of the symmetries we have noted, the P(e) expression (3.1-7) can be modified to

$$P(e) = \frac{1}{4}\{P(e|010) + 2P(e|011) + P(e|111)\}. \tag{3.1-20}$$

### 3.1.1.2 Numerical Results for No Jamming

The average BER (3.1-20) is computed using the program given in Appendix B, with the results as shown in Figures 3.1-1 to 3.1-3.

In Figure 3.1-1, the time-bandwidth product $D = W_{IF}T$ is fixed at the value of $D = 1$, while the digital FM modulation index, h, is varied. The well-known property that $h = 0.7$ is the best value for the additive Gaussian channel with high SNR is illustrated by the figure. Crossovers in the figure reveal that this accepted value of h is best for SNR's greater than 4-5 dB; for lower SNR, $h = 0.8$ is the best value among the values of this parameter which are plotted.

In Figure 3.1-2, the modulation index is held fixed at the value $h = 0.7$, while D is varied. Plotted against SNR, as in this figure, the BER decreases uniformly as the filter bandwidth increases (D increases). However, this is a misleading portrayal of the dependence of the BER on D, since the SNR is given by

$$\mathrm{SNR} = \frac{S}{N} = \frac{ST}{N_0 W_{IF} T} = \frac{E_b}{N_0 D} \quad ; \tag{3.1-21}$$

in order for SNR to remain constant while D varies, the bit-energy-to-noise density ratio must vary. A more fair, equal-bit-energy comparison is presented in Figure 3.1-3, in which the BER is plotted against $E_b/N_0$. This second comparison is more in accord with intuition, for we observe a tradeoff between

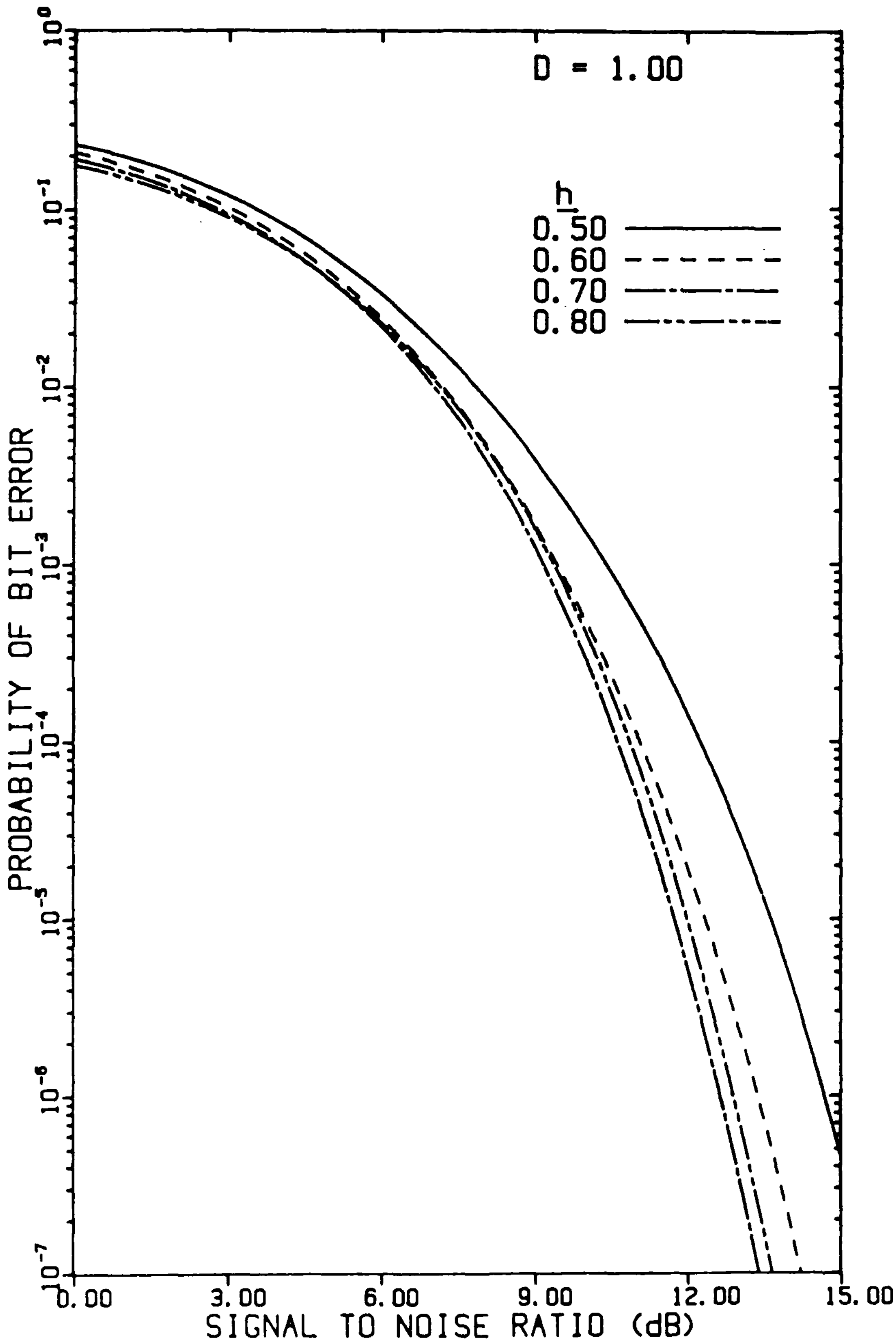



FIGURE 3.1-1 UNJAMMED CPFSK BER VS. SNR FOR D = 1, WITH MODULATION INDEX VARIED

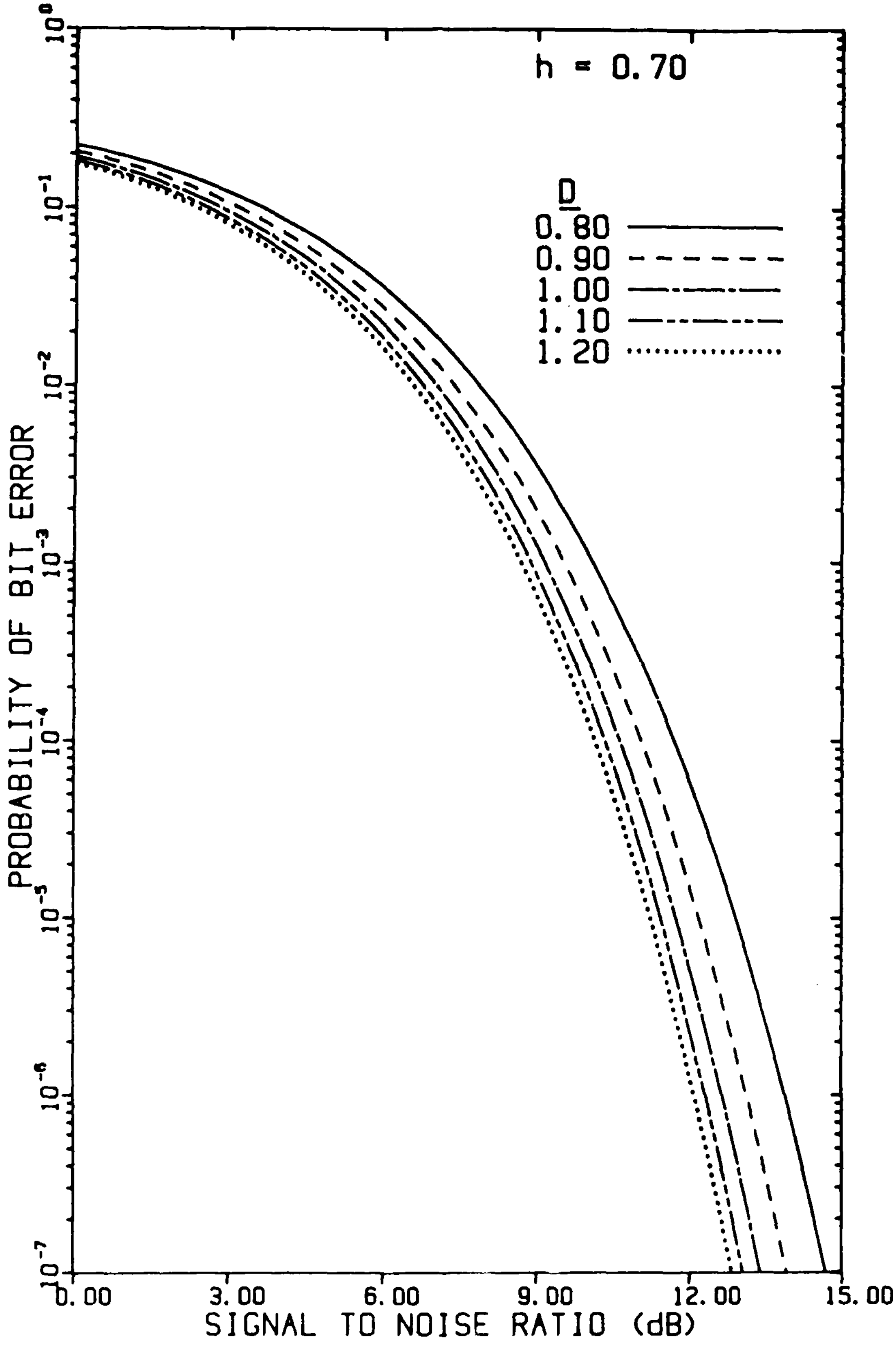



FIGURE 3.1-2 UNJAMMED CPFSK BER VS. SNR FOR h = 0.7, WITH TIME-BANDWIDTH PRODUCT VARIED

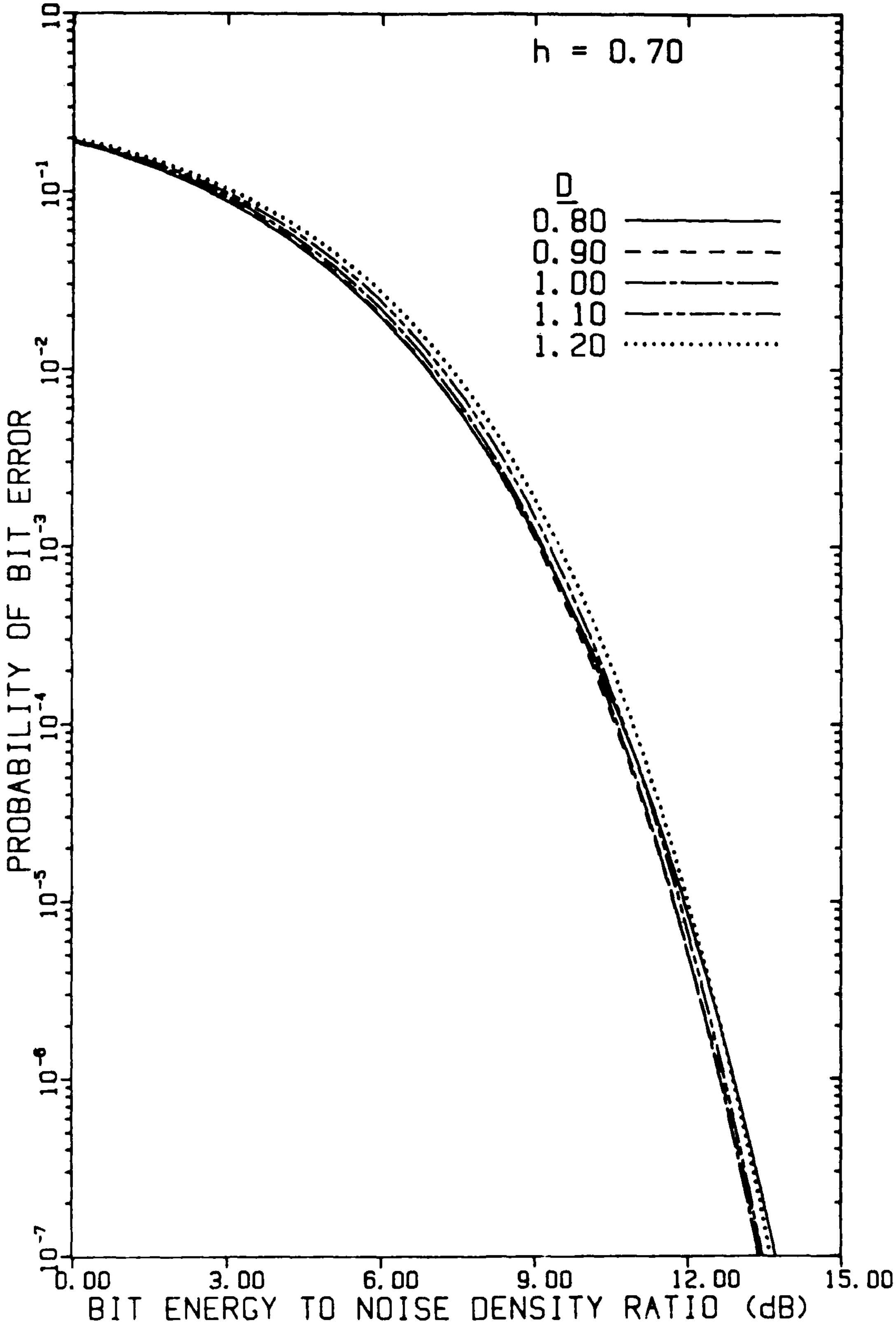



FIGURE 3.1-3 UNJAMMED CPFSK BER VS. $E_b/N_0$ FOR h = 0.7, WITH TIME-BANDWIDTH PRODUCT VARIED.

the beneficial effects of increasing D (more signal energy recovered) and the degrading effects (more noise power admitted), with D = 1.0 being optimum for high $E_b/N_0$, D = 0.9 best for $8 < E_b/N_0 < 12$ dB, and D = 0.8 best for $E_b/N_0 < 8$ dB, among the values of D used in the figure.

### 3.1.2 Case of Partial-Band Jamming

The computations for no jamming are parametric in the SNR, yielding

$$P(e) = P(e;\rho_N). \tag{3.1-22}$$

Inclusion of partial-band jamming is performed by calculating the expression

$$P(e) = (1-\gamma)\, P(e;\rho_N) + \gamma P(e;\rho_T), \tag{3.1-23}$$

where

$$\rho_T = \frac{\gamma\rho_J\rho_N}{\rho_N+\gamma\rho_J} \tag{3.1-24}$$

is the effective SNR when the signal is jammed, and $\gamma$ is the fraction of the hopping band which is jammed.

Figure 3.1-4 shows plots of the jammed BER (3.1-23) as a function of $E_b/N_J = \rho_J$, and parametric in $\gamma$, for $E_b/N_0$ fixed at 11.75 dB. This value of $E_b/N_0$ gives a $10^{-5}$ BER without jamming, as can be observed in the figure as $E_b/N_J$ becomes large. For each value of $\gamma$, the BER curve is "S-shaped," with a left (low $E_b/N_J$ or strong jamming) asymptote close to the value of $P(e) = \gamma/2$, and with a right asymptote (high $E_b/N_J$ or weak jamming) of $P(e) = 10^{-5}$.

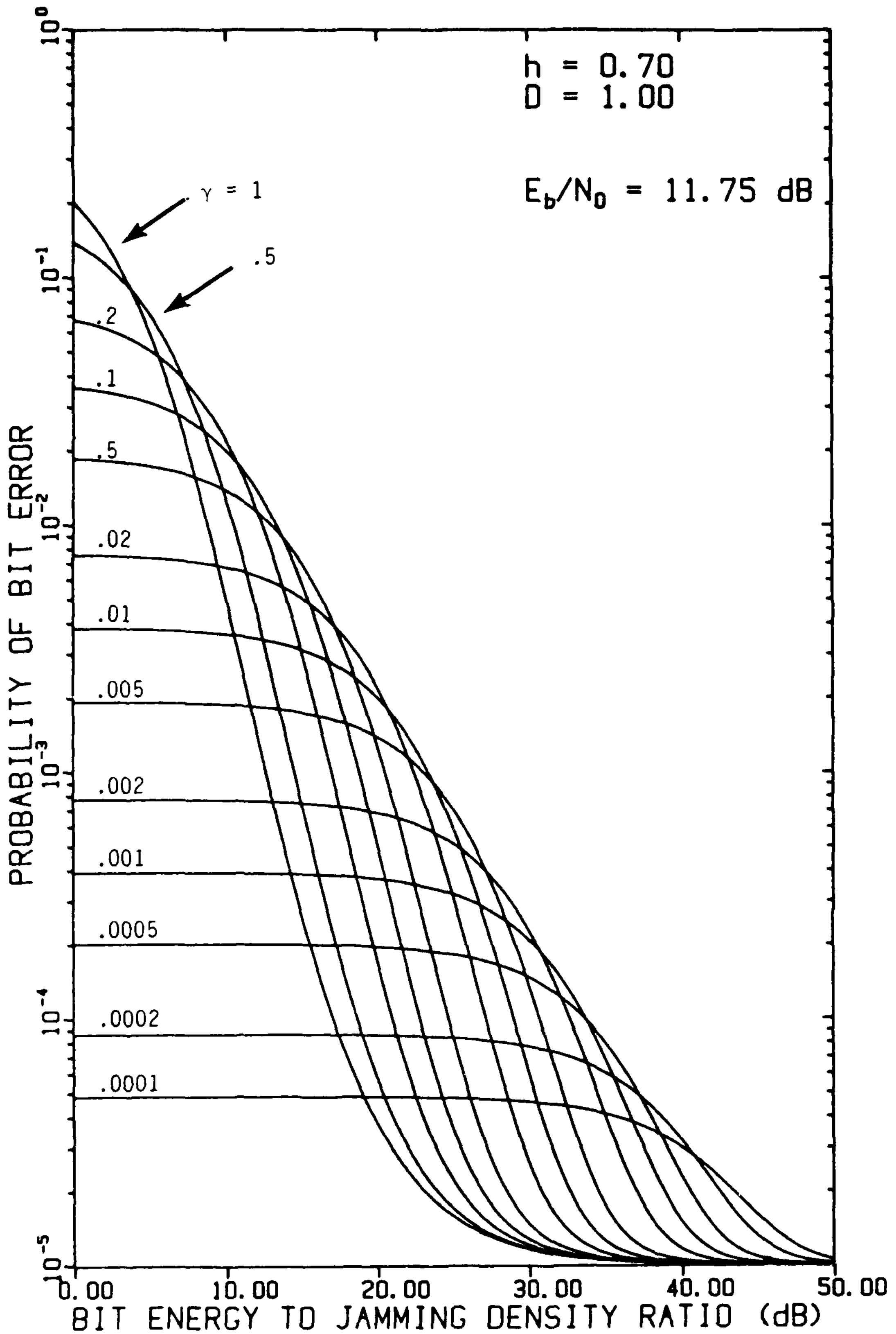



FIGURE 3.1-4 FH/CPFSK BER VS. $E_b/N_J$ IN PARTIAL-BAND JAMMING, FOR ONE HOP/BIT AND $E_b/N_0$ = 11.75 dB (GIVING A $10^{-5}$ BER WITHOUT JAMMING), PARAMETRIC IN $\gamma$, THE FRACTION OF BAND JAMMED.

The centers of the "S-curves" are different for each value of $\gamma$, so that when they are superimposed as in the figure, it is evident that a particular value of $\gamma$ produces the worst-case or highest BER value only for a small range of $E_b/N_J$. For example, the value $\gamma = 0.01$ is the worst-case, among the values displayed, for 21 dB < $E_b/N_J$ < 24 dB, approximately.

Note how much greater the error is with worst-case jamming than with full-band jamming ($\gamma = 1$). For example, $10^{-4}$ is achieved when $\gamma = 1$ for $E_b/N_J$ = 17 dB; when $\gamma$ is the worst-case value, a $10^{-4}$ BER requires $E_b/N_J$ = 34 dB, or 17 dB more signal power.

If we were to superimpose additional curves for more closely-spaced values, it is easy to predict that the appearance of Figure 3.1-4 would become essentially that of a black solid, with its upper "edge" a plot of the worst-case BER for $E_b/N_J$ continuously varied, or

$$P_{WC} = \max_{\gamma} P(e;E_b/N_J). \tag{3.1-25}$$

Figure 3.1-4 suggests that the slope of that worst-case BER curve would be -1, that is,

$$P_{WC} = \frac{0.21}{E_b/N_J}\,,\ 2\text{ dB} < E_b/N_J < 35\text{ dB} \tag{3.1-26}$$

for much of the range of $E_b/N_J$, with the slope eventually increasing to zero as $E_b/N_J$ increases beyond 35 dB or so, since the BER cannot be less than the unjammed value of $10^{-5}$.

If $E_b/N_0$ is increased to 15 dB, giving an unjammed BER << $10^{-7}$ (see Figure 3.1-1), essentially the jammed BER is given by

$$P(e) \approx \gamma P(e;\rho_T), \tag{3.1-27}$$

and the superimposed fixed-$\gamma$ curves are as shown in Figure 3.1-5. They are still "S-curves" as in Figure 3.1-4, but the high $E_b/N_J$ asymptote is much lower, and the low $E_b/N_J$ asymptotes are somewhat lower. Now we see that the worst-case BER is characterized by a -1 slope throughout the range of $E_b/N_J$ shown, so that

$$P_{WC} = \frac{0.21}{E_b/N_J}, \quad E_b/N_J > 2 \text{ dB}, \tag{3.1-28a}$$

with the worst-case value of $\gamma$ being very closely predicted by

$$\gamma_{WC} = \begin{cases} \dfrac{1.58}{E_b/N_J}, & E_b/N_J > 1.58 = 2 \text{ dB} \\ 1, & E_b/N_J < 1.58 = 2 \text{ dB}. \end{cases} \tag{3.1-28b}$$

### 3.1.3 Simplified Calculations for L = 1

It shall be interesting to compare the exact BER results for L = 1, presented above, with calculations based on simplifying assumptions. The expressions (2.1-55) and (2.1-56) for the differential phase distribution's pdf and probability function simplify considerably for the following assumptions:

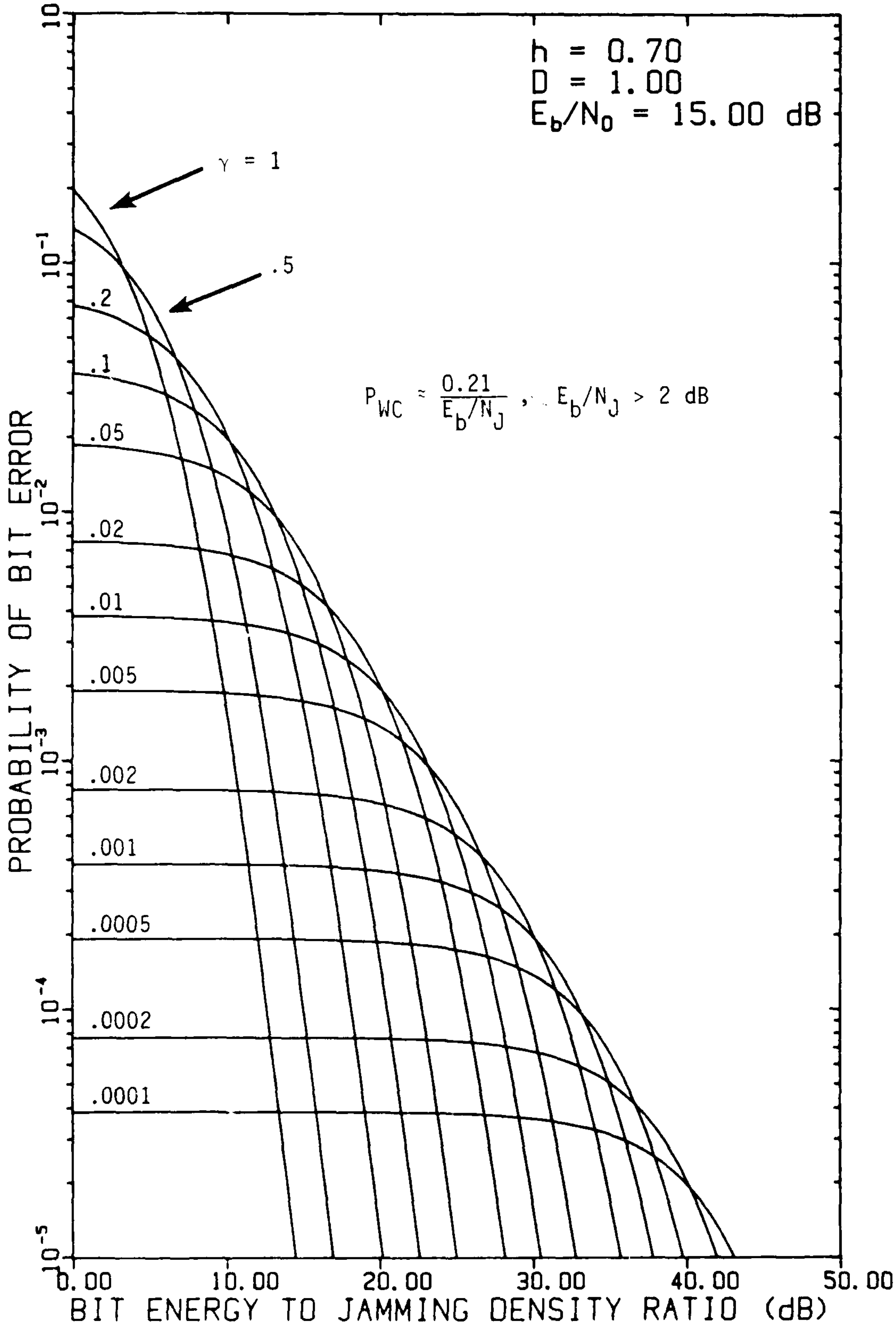



FIGURE 3.1-5 FH/CPFSK BER VS. $E_b/N_J$ IN PARTIAL-BAND JAMMING FOR ONE HOP/BIT AND $E_b/N_0$ = 15 dB, PARAMETRIC IN γ, THE FRACTION OF BAND JAMMED

$$\Delta\phi = \pi/2 \quad \text{(representative or typical value)}$$

$$r = \lambda = 0 \quad \text{(neglecting correlations)}$$

$$\left.\begin{aligned} V &= 0 \\ U &= W = \rho\, a^2 \end{aligned}\right\} \quad \text{(neglecting intersymbol interference asymmetries).} \tag{3.1-29}$$

In effect we are replacing the distribution, averaged over the pattern-dependent quantities, with a distribution with "typical" values. In that the $V = 0$ assumption relates to the "all one's" pattern, it is reasonable to interpret $\Delta\phi = \pi/2$ as arising from a modulation index value of $h = 0.5$, so that $a^2 = 0.81 \approx \exp(-\pi/16)$ is appropriate. However, the analysis approach used here is not so formal as to prevent an arbitrary choice of some value for $a^2$, if it turns out to model the exact results well.

The jammed BER for the simplifying assumptions becomes

$$P(e) = (1-\gamma)\left[c_0 e^{-c_0} + 1 - e^{-c_0}\right] + \gamma\left[c_1 e^{-c_1} + 1 - e^{-c_1}\right], \tag{3.1-30a}$$

where

$$c_i = \frac{1}{4}\, e^{-a^2 \rho_i} = P(e|\text{no clicks}, i), \tag{3.1-30b}$$

and

$$\rho_i = \begin{cases} \rho_N, & i = 0 \quad \text{(unjammed)} \\ \rho_T, & i = 1 \quad \text{(jammed).} \end{cases} \tag{3.1-30c}$$

By coincidence, the probability of zero clicks turns out to be $e^{-c_i}$ as shown; that is, if the phase modulation is assumed to be such that $\dot{\phi} = \pi h/T = \pi/2T$.

For no jamming ($\gamma = 0$), the simplified BER expression using $a^2 = 0.81$ gives the following values:

| $\epsilon_N$ | BER |
|---|---|
| 10 dB | 1.5176 (-4) |
| 11.25733 dB | 1.0000 (-5) |
| 15 dB | 1.8781 (-12) . |

Calculation of the worst-case partial-band jamming BER produces the following asymptotic results (higher $E_b/N_J$):

| $\epsilon_N$ | $P_{WC}$ | $\gamma_{WC}$ | |
|---|---|---|---|
| 11.25733 dB | $0.23/(E_b/N_J)$ | $1.7/(E_b/N_J)$ | (3.1-31a) |
| 15 dB | $0.22/(E_b/N_J)$ | $1.4/(E_b/N_J)$. | (3.1-31b) |

Note that this approximation gives a higher jammed error than the exact expression [see (3.1-28)], and a lower unjammed error. Thus, "tinkering" with the vlaue of $a^2$ (to scale the SNR) will not accomplish agreement between approximate and exact answers for both unjammed and jammed conditions.

Another approximation approach is to fit a simplified curve to the exact unjammed P(e) shown in, for example, Figure 3.1-1. For $h = 0.7$ and $D = 1.0$, we have the approximation

$$P(e) \approx 0.394\, e^{-0.717\rho} , \tag{3.1-32}$$

which is fitted precisely at $\rho = 0$ dB and 10 dB; it gives a $10^{-5}$ BER for 11.7 dB. When this form is used in the jammed BER equation, for high $E_b/N_0$ (negligible thermal noise), the worst-case BER obtained is

$$P_{WC} \approx \frac{.20}{E_b/N_J} , \tag{3.1-33a}$$

with

$$\rho_{WC} \approx \frac{1.39}{E_b/N_J} . \tag{3.1-33b}$$

These results compare well with the exact case given in (3.1-28).

## 3.2 BER COMPUTATIONS FOR L = 2

The calculations of the BER for L = 1 showed that worst-case partial-band jamming severely degrades the FH/CPFSK performance. We now begin to determine whether linear diversity combining will produce an improvement.

### 3.2.1 Derivation of BER Expressions

The two differential phase samples from two hops for a particular bit are statistically independent, due to the separation in time and frequency of the noises added to the signal on the two hops. Therefore, the joint pdf for the two samples is, for a particular data pattern,

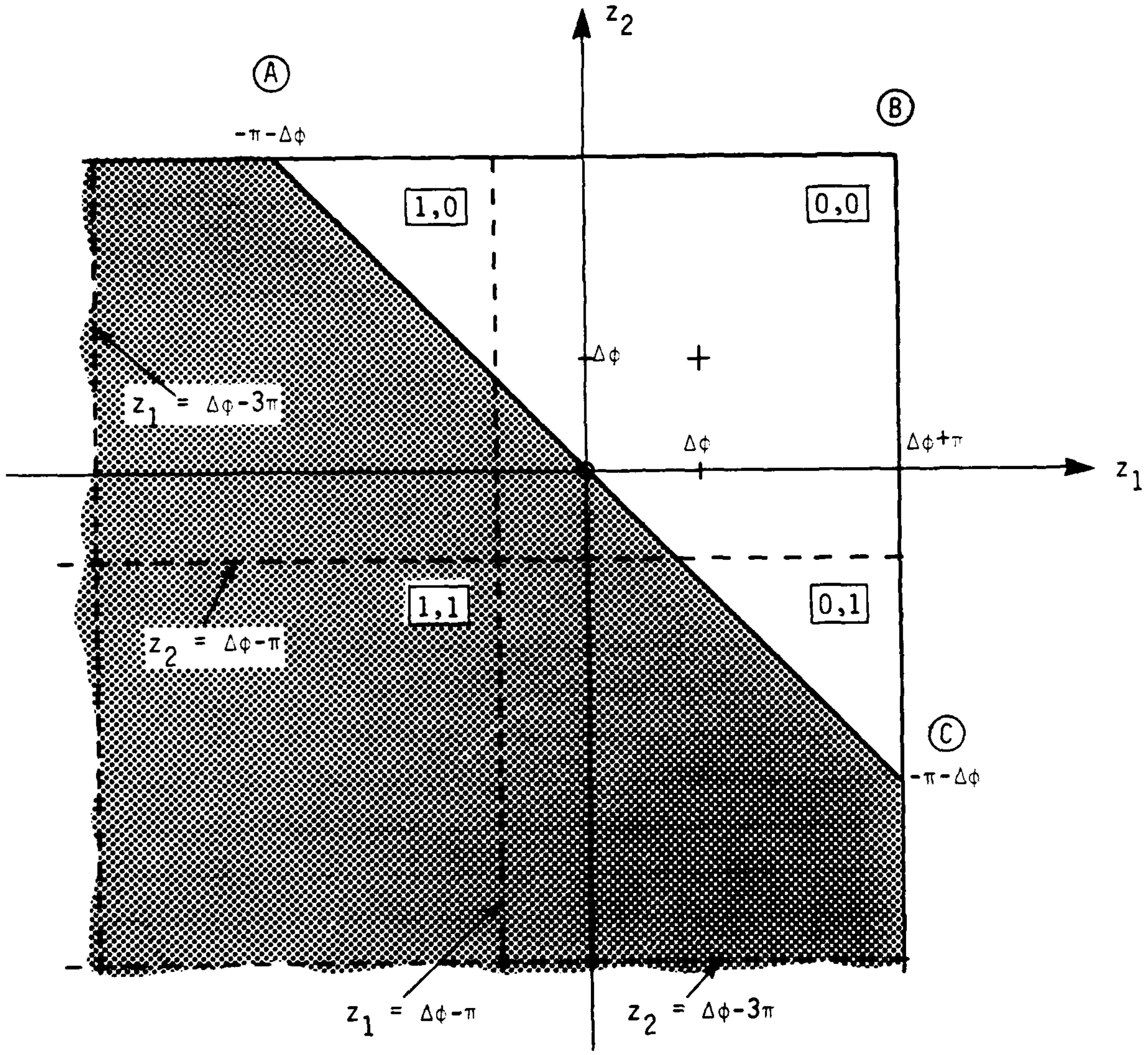
$$p_{z_1,z_2}(x,y|\underline{\varepsilon}) = p^{(1)}(x|\underline{\varepsilon})\, p^{(1)}(y|\underline{\varepsilon}), \qquad (3.2\text{-}1)$$

using the notation of (3.1-2).

Restricting our attention to patterns with a bit value of 1, the conditional probability of error is given by

$$P(e|\underline{\varepsilon}) = \Pr\{z_1+z_2 < 0|\underline{\varepsilon}\}, \qquad (3.2\text{-}2)$$

and the probability of error is represented by the area noted in Figure 3.2-1. Since the joint pdf (3.2-1) is symmetric about the line $z_1 = z_2$, the conditional BER, with the help of the figure, is seen to be




Shaded area represents P(e)

Area (A)(B)(C) represents 1-P(e)

Number of clicks: $\boxed{N_{c_1}, N_{c_2}}$

FIGURE 3.2-1 AREAS REPRESENTING THE BER FOR L = 2

$$P(e|\underline{\beta}) = 1 - \int_{-\pi-\Delta\phi}^{\pi+\Delta\phi} dx \int_{-x}^{\pi+\Delta\phi} dy\; p^{(1)}(x|\underline{\beta})\; p^{(1)}(y|\underline{\beta}), \tag{3.2-3a}$$

or

$$1 - P(e|\underline{\beta}) = 2\int_{-\pi-\Delta\phi}^{\Delta\phi-\pi} dx \int_{-x}^{\pi+\Delta\phi} dy\; p^{(1)}(x|\underline{\beta})\; p^{(1)}(y|\underline{\beta})$$

$$+ \int_{\Delta\phi-\pi}^{\Delta\phi+\pi} dx \int_{\max(-x,\Delta\phi-\pi)}^{\Delta\phi+\pi} dy\; p^{(1)}(x|\underline{\beta})\; p^{(1)}(y|\underline{\beta}) \tag{3.2-3b}$$

$$\equiv \quad 2\,P_1 + P_2. \tag{3.2-3c}$$

Recall that

$$p^{(1)}(x,\underline{\beta}) = \sum_{n=0}^{\infty} f_n\,(x+2\pi n;\underline{\beta})$$

$$= (1-\gamma) \sum_{n=0}^{\infty} p_{0,n} p_{\psi}(x+2\pi n;\rho_N,\underline{\beta})$$

$$+ \gamma \sum_{n=0}^{\infty} p_{1,n} p_{\psi}(x+2\pi n;\rho_T,\underline{\beta}), \tag{3.2-4}$$

where the modulo $2\pi$ pdf $p_{\psi}(x)$ is nonzero for $|x-\Delta\phi| < \pi$, and is given by (2.1-55). For notational convenience, let

$$p_{\psi}(x;\rho,\underline{\beta}) = \begin{cases} q_0(x), & \rho = \rho_N \text{ (unjammed)} \\ q_1(x), & \rho = \rho_T \text{ (jammed)}\,. \end{cases} \tag{3.2-5}$$

With this notation, we have

$$P_1 = \int_{-\pi-\Delta\phi}^{\Delta\phi-\pi} dx \int_{-x}^{\pi+\Delta\phi} dy \left[ (1-\gamma)p_{0,1}\, q_0(x+2\pi) + \gamma p_{1,1}\, q_1(x+2\pi)\right]$$

$$\times \left[(1-\gamma)p_{0,0}\, q_0(y) + \gamma p_{1,0}\, q_1(y)\right]$$

$$= \int_{\pi-\Delta\phi}^{\Delta\phi+\pi} dx \int_{2\pi-x}^{\Delta\phi+\pi} dy \Big[ (1-\gamma)^2\, p_{0,0}\, p_{0,1}\, q_0(x)\, q_0(y)$$

$$+ \gamma(1-\gamma)p_{0,0}\, p_{1,1}\, q_1(x)\, q_0(y)$$

$$+ \gamma(1-\gamma)\, p_{1,0}\, p_{0,1}\, q_0(x)\, q_1(y)$$

$$+ \gamma^2 p_{1,1}\, p_{1,0}\, q_1(x)\, q_1(y)\Big]$$

$$= (1-\gamma)^2\, p_{0,0}p_{0,1}\, P_1(0,0) + \gamma(1-\gamma)\, p_{0,0}\, p_{1,1}\, P_1(1,0)$$

$$+ \gamma(1-\gamma)\, p_{1,0}\, p_{0,1}\, P_1(0,1) + \gamma^2 p_{1,1}p_{1,0}\, P_1(1,1). \qquad (3.2\text{-}6)$$

From (2.1-56), we recall that

$$\int_A^B dy\; q_i(y) = F_i(B) - F_i(A) + \frac{1}{2}\left[\mathrm{sgn}\,(B-\Delta\phi) - \mathrm{sgn}(A-\Delta\phi)\right], \qquad (3.2.7a)$$

using the notation

$$F(\cdot;\rho,\underline{E}) = \begin{cases} F_0(\cdot), & \rho = \rho_N \text{ (unjammed)} \\ F_1(\cdot), & \rho = \rho_T \text{ (jammed)}, \end{cases} \qquad (3.2\text{-}7b)$$

and realizing that

$$F_i(x \pm 2k\pi) = F_i(x). \tag{3.2-7c}$$

Substituting the defined functions results in

$$\begin{aligned} P_1(i,j|\underline{\varepsilon}) = {} & F_j(\Delta\phi+\pi)\,[F_i(\Delta\phi+\pi) - F_i(\pi-\Delta\phi)] \\ & - \int_{\pi-\Delta\phi}^{\Delta\phi+\pi} dx\; q_i(x)\; F_j(-x) \\ & + u(\Delta\phi-\pi/2)[F_j(\Delta\phi+\pi) + F_i(\Delta\phi+\pi) - F_i(2\pi-\Delta\phi)], \end{aligned} \tag{3.2-8}$$

in which the last term is zero for $\Delta\phi < \pi/2$.

The term $P_2$ may be written

$$\begin{aligned} P_2 = {} & [(1-\gamma)p_{0,0} + \gamma p_{1,0}]^2 \\ & - \int_{\Delta\phi-\pi}^{\pi-\Delta\phi} dx \int_{\Delta\phi-\pi}^{-x} dy\; [(1-\gamma)\, p_{0,0}\, q_0(x) + \gamma\, p_{1,0}\, q_1(x)] \\ & \times [(1-\gamma)p_{0,0}\, q_0(y) + \gamma p_{1,0}\, q_1(y)] \\ = {} & [(1-\gamma)p_{0,0} + \gamma p_{1,0}]^2 - (1-\gamma)^2 p_{0,0}^2\, P_2(0,0) \\ & - \gamma(1-\gamma)p_{0,0}p_{1,0}[P_2(0,1) + P_2(1,0)] - \gamma^2 p_{1,0}^2 P_2(1,1). \end{aligned} \tag{3.2-9}$$

The probabilities $P_2(i,j)$ are found to be

$$
\begin{aligned}
P_2(i,j|\underline{\beta}) = & \int_{\Delta\phi-\pi}^{\pi-\Delta\phi} dx\, q_i(x) F_j(-x) \\
& - F_j(\Delta\phi-\pi)\,[F_i(\pi-\Delta\phi) - F_i(\Delta\phi-\pi)] \\
& + u(\pi/2-\Delta\phi)[F_i(-\Delta\phi) - F_i(\Delta\phi-\pi) - F_j(\Delta\phi-\pi)]
\end{aligned}
\qquad (3.2\text{-}10)
$$

where the last term is zero for $\Delta\phi > \pi/2$.

The total P(e) then can be expressed as

$$
P(e|\underline{\beta}) = \sum_{i=0}^{1} \sum_{j=0}^{1} (1-\gamma)^{2-(i+j)}\, \gamma^{i+j}\, P(e;i,j|\underline{\beta}) \qquad (3.2\text{-}11a)
$$

where

$$
\begin{aligned}
P(e;i,j|\underline{\beta}) = & \; 1 - p_{i,0}\, p_{j,0} - 2p_{i,1}\, p_{j,0}\, P_1(i,j|\underline{\beta}) \\
& + p_{i,0}\, p_{j,0}\, P_2(i,j|\underline{\beta}).
\end{aligned}
\qquad (3.2\text{-}11b)
$$

For computation, it is somewhat more efficient to use the following formulation:

$$
P(e|\underline{\beta}) = \sum_{i=0}^{1} \sum_{j=0}^{i} \binom{2}{i+j} (1-\gamma)^{2-(i+j)}\, \gamma^{i+j}\, P_3(i,j|\underline{\beta}) \qquad (3.2\text{-}12a)
$$

where

$$P_3(i,j|\underline{\beta}) = 1 - p_{i,0}\, p_{j,0} - p_{i,1}\, p_{j,0}\, P_1(i,j|\underline{\beta}) - p_{i,0} p_{j,1} P_1(j,i|\underline{\beta})$$

$$+ p_{i,0}\, p_{j,0}\, P_2(i,j|\underline{\beta}). \qquad (3.2\text{-}12b)$$

### 3.2.2 BER Results for L = 2

The L = 2 bit error probability for FH/CPFSK in partial-band noise jamming was calculated as the average over bit patterns:

$$P(e) = \frac{1}{4}\,[P(e|111) + 2P(e|011) + P(e|010)], \qquad (3.2\text{-}13)$$

where the pattern-dependent error probability, parametric in $\gamma$, the jamming fraction, is given by (3.2-12). The program listed in Appendix C was used.

The results in Figure 3.2-2 demonstrate that the BER for L = 2 is maximized for a particular value of $\gamma$ which depends on $E_b/N_J$. For the case of $E_b/N_0$ = 15 dB, h = 0.7, and D = 1.0 as shown in the figure, we can observe that the worst-case $\gamma$ value is approximately

$$\gamma_{WC} \approx \begin{cases} 4/(E_b/N_J) & , \ E_b/N_J > 4 = 6 \text{ dB}; \\ 1 & , \ E_b/N_J < 6 \text{ dB}; \end{cases} \qquad (3.2\text{-}14)$$

for high $E_b/N_0$. From Figure 3.2-2 we can develop the plots of BER vs. $E_b/N_J$ shown in Figure 3.2-3, for fullband jamming ($\gamma$ = 1) and for worst-case partial-band jamming ($\gamma = \gamma_{WC}$). It is clear that the jamming is significantly more effective using $\gamma = \gamma_{WC}$ than using $\gamma = 1$, when $E_b/N_J > 6$ dB. For example,

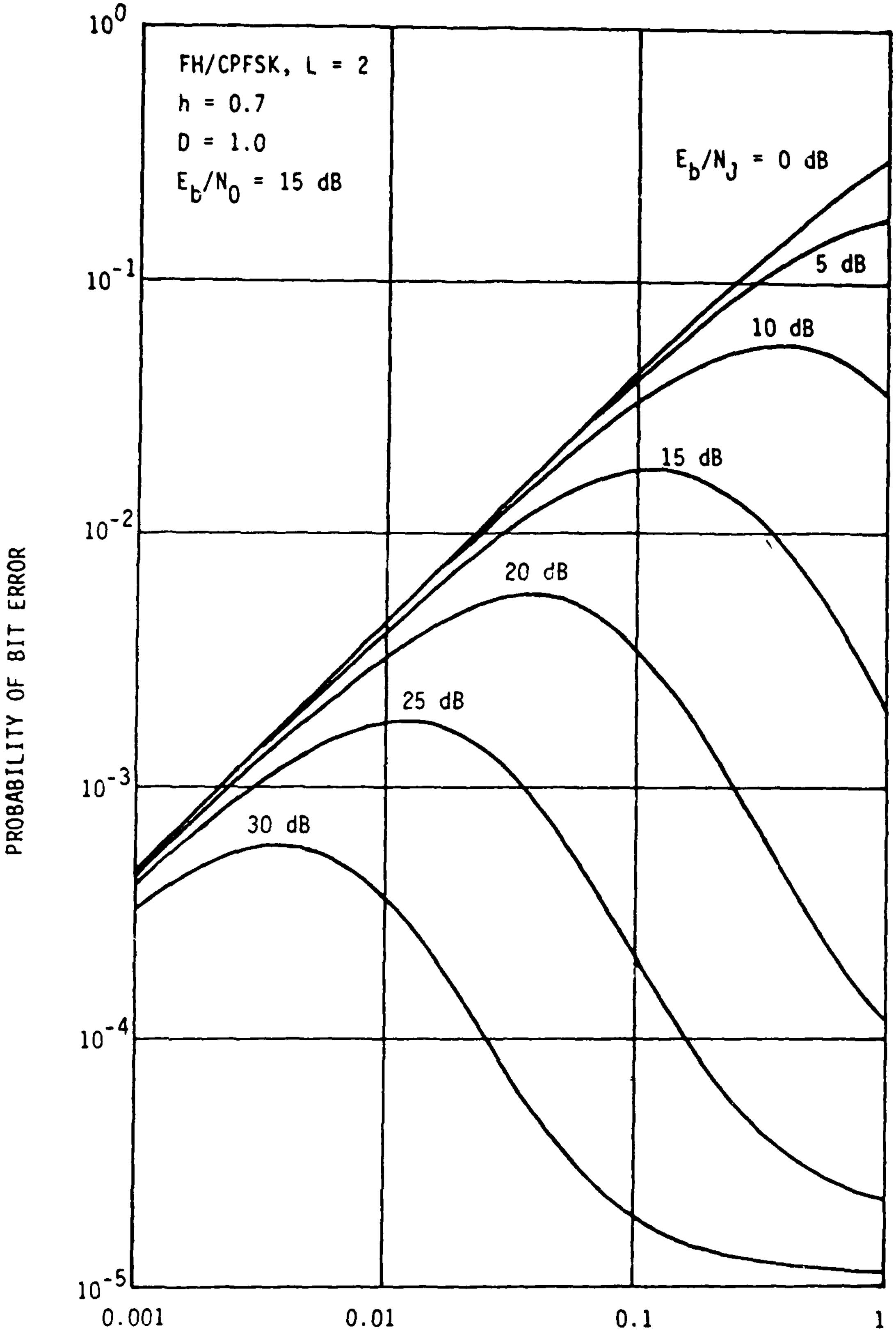



FIGURE 3.2-2 FH/CPFSK BER VS. $\gamma$ FOR L = 2 HOPS/BIT, WHEN $E_b/N_0$ = 15 dB AND PARAMETRIC IN $E_b/N_J$

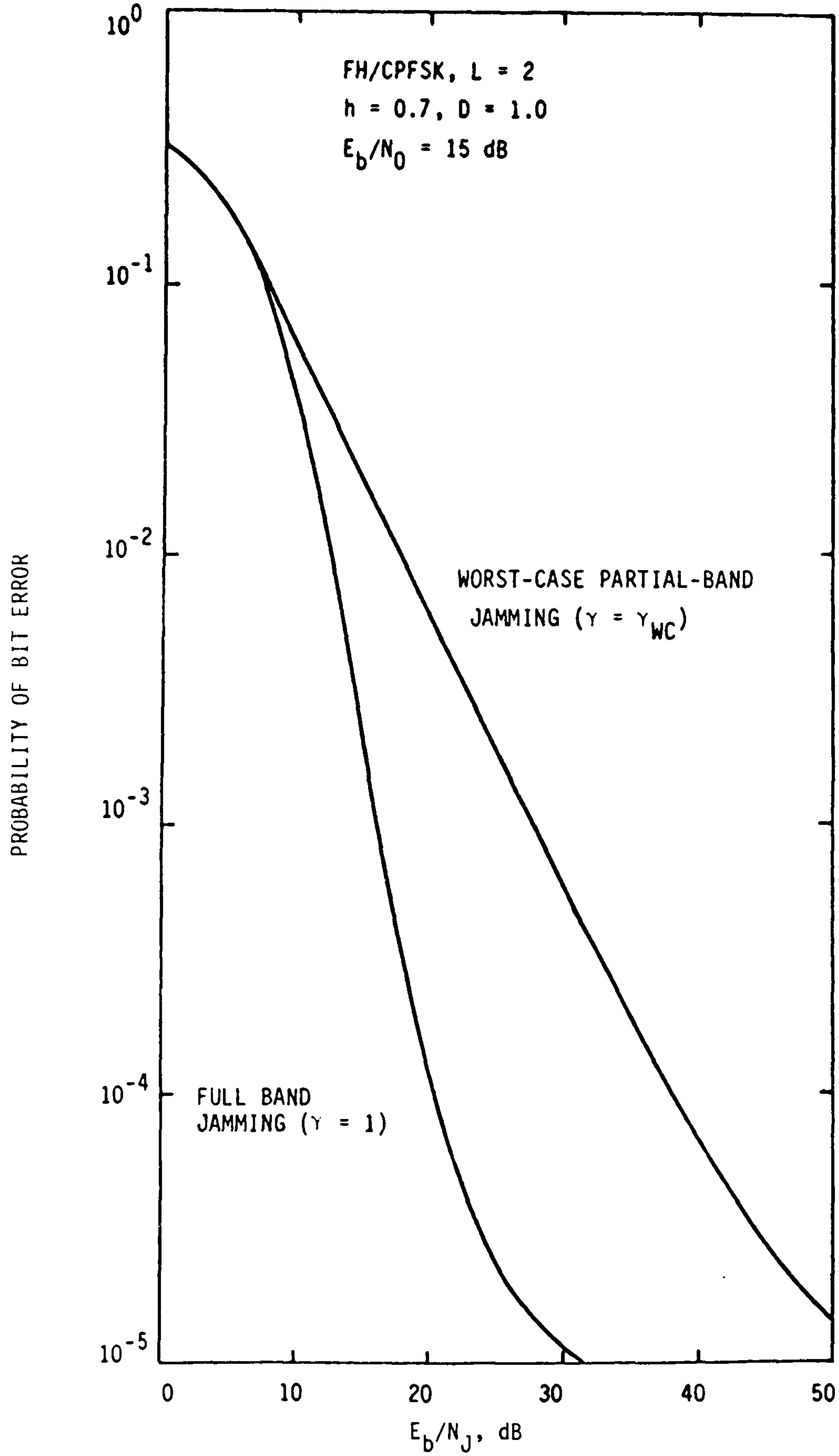



FIGURE 3.2-3 FH/CPFSK BER VS. $E_b/N_J$ for L = 2 HOPS/BIT, WHEN $E_b/N_0$ = 15 dB FOR FULLBAND JAMMING ($\gamma$ = 1) AND FOR WORST-CASE PARTIAL-BAND JAMMING.

a $10^{-4}$ BER requires $E_b/N_J$ = 20.5 dB when $\gamma = 1$, but $E_b/N_J$ = 37.6 dB when $\gamma = \gamma_{WC}$, a 17.1 dB difference. We note that the worst-case BER tends to exhibit an inverse-linear dependence upon $E_b/N_J$ when $E_b/N_J > 6$ dB; that is,

$$P_{WC} \simeq .58/(E_b/N_J), \quad E_b/N_J > 6 \text{ dB}, \tag{3.2-15}$$

for high $E_b/N_0$. This BER is averaged over the data patterns; it is interesting to note that the worst-case BER for each pattern also tends to have the same kind of dependence upon $E_b/N_J$, with the coefficients being approximately .74 for 111, .58 for 011, and .42 for 010. Thus an analysis based on only the "middle" case of 011 would have actually represented the average in this instance.

Recalling that $P_{WC} \simeq .21/(E_b/N_J)$ for L = 1, we observe that the linear diversity combining of two differential phase samples does not improve the system's performance, but rather degrades it by about 4.4 dB. This difference is partly understood from the fact that, under a bit energy constraint, $\xi_N = \frac{1}{2}(E_b/N_0)$ and $\xi_T = \frac{1}{2}(E_b/N_T)$ for L = 2. But this fact only accounts for a 3 dB difference. The entire 4.4 dB difference may be attributed to noncoherent combining losses, which evidently are so great for linear combining of CPFSK differential phase samples that even under an equal power constraint ($\xi_N = E_b/N_0$, etc.), L = 2 gives a worse result than L = 1.

For example, with no jamming, $E_b/N_0$ = 15 dB yields a BER of $1.83 \times 10^{-10}$ for L = 1, and $8.3 \times 10^{-6}$ for L = 2; this difference in performance is equivalent to about 3.2 dB in $E_b/N_0$. So, even without jamming, the L = 2 combining losses are great enough to give a worse BER result for an equal power constraint.

### 3.2.3 Simplified Calculations

As in Section 3.1.3 for L = 1, we now summarize L = 2 BER results using the "typical pattern" approach, rather than the exact one involving averaging over patterns. The simplifying case is that of $\Delta\phi = \pi/2$, $r = \lambda = 0$, and $V = 0$, which gives

$$P(e;i,j) = 1 - p_{i,0}\, p_{j,0} + 2\, p_{i1} p_{j,0} \int_{\pi/2}^{3\pi/2} dx\, q_i(x) F_j(-x)$$

$$+ p_{i,0}\, p_{j,0} \int_{-\pi/2}^{\pi/2} dx\, q_i(x) F_j(-x), \tag{3.2-16a}$$

where

$$q_i(x) = \frac{1}{4\pi} \int_{-\pi/2}^{\pi/2} dy \cos y\, (1 + U_i + U_i \cos y \sin x) e^{-U_i(1 - \cos y \sin x)} \tag{3.2-16b}$$

and

$$F_j(-x) = \frac{\cos x}{4\pi} \int_{-\pi/2}^{\pi/2} dy\, \frac{e^{-U_j(1 + \cos y \sin x)}}{1 + \cos y \sin x} \quad . \tag{3.2-16c}$$

Using these expressions with $U_i = \varepsilon_i a^2/L = 0.405\, \varepsilon_i$ and the click probability formulas in (3.1-30) gives a worst-case BER of

$$P_{WC} \approx \frac{.50}{E_b/N_J} \tag{3.2-17}$$

for high $E_b/N_J$, in the linear portion of the error curve. It is interesting that this L = 2 "typical" BER is better than the exact L = 2 performance, whereas for L = 1 the typical BER is worse, but no explanation is immediately apparent.

### 3.3 BER COMPUTATIONS FOR L > 2

The intricacy of the FH/CPFSK BER calculation was seen in the last subsection to be considerable for L = 2, with corresponding amounts of computer time needed. In the following material we summarize the extension to L > 2 of the direct, or convolutional, analysis approach used previously. We also introduce another numerical approach which is designed to control the amount of computation required to evaluate the BER. Finally, numerical results for worst-case BER when L > 2 are presented which indicate the failure of linear diversity combining to achieve an improvement in the system performance against worst-case partial-band jamming.

#### 3.3.1 Methodology Using Direct Approach

Assuming that the pdf for the sum of L-1 diversity samples can be calculated, given by

$$p^{(L-1)}(x|\underline{\varepsilon}) = \sum_{n=0}^{\infty} f_n^{(L-1)}(x+2\pi n;\underline{\beta}) \tag{3.3-1}$$

in the notation of Section 3.1, we can formulate the BER for the sum of L samples in terms of this pdf and the L = 1 pdf. It can be shown that

$$P(e|\underline{\varepsilon}) = 1 - \sum_{n=0}^{nmax} \sum_{m=0}^{n} B_{m,n-m}^{(L)} \quad , \tag{3.3-2}$$

where nmax is the number of significant clicks, given by (2.2-22b), and $B_{m,n-m}^{(L)}$ is the probability that the sum is positive when there are m clicks in the Lth sample and n-m clicks in the sum of the first L-1 samples.

That is,

$$B_{m,n-m}^{(L)} = \int_{\max[\Delta\phi-\pi-2\pi m,-(L-1)(\Delta\phi+\pi)+2\pi(n-m)]}^{\Delta\phi+\pi-2\pi m} dx\ f_m^{(1)}(x+2\pi m) \int_{\max[(L-1)(\Delta\phi-\pi)-2\pi(n-m),-x]}^{(L-1)(\Delta\phi+\pi)-2\pi(n-m)} dy\ f_{n-m}^{(L-1)}[y+2\pi(n-m)]$$

$$= \int_{\max[\Delta\phi-\pi,2\pi n-(L-1)(\Delta\phi+\pi)]}^{\Delta\phi+\pi} dx\ f_m^{(1)}(x) \int_{\max[(L-1)(\Delta\phi-\pi),-x+2\pi n]}^{(L-1)(\Delta\phi+\pi)} dy\ f_{n-m}^{(L-1)}(y). \qquad (3.3\text{-}3)$$

For example, if $L = 3$ and $\Delta\phi > \pi/3$,

$$P(e|\underline{\beta}) = 1 - B_{00} - (B_{01}+B_{10}) - (B_{02}+B_{11}+B_{20}), \qquad (3.3\text{-}4)$$

where

$$B_{00} = \int_{\Delta\phi-\pi}^{\Delta\phi+\pi} dx\ f_0^{(1)}(x) \int_{\max[2(\Delta\phi-\pi),-x]}^{2(\Delta\phi+\pi)} dy\ f_0^{(2)}(y) \qquad (3.3\text{-}5a)$$

$$B_{01} = \int_{\Delta\phi-\pi}^{\Delta\phi+\pi} dx\ f_0^{(1)}(x) \int_{-x+2\pi}^{2(\Delta\phi+\pi)} dy\ f_1^{(2)}(y) \qquad (3.3\text{-}5b)$$

$$B_{10} = \int_{\Delta\phi-\pi}^{\Delta\phi+\pi} dx\ f_1^{(1)}(x) \int_{-x+2\pi}^{2(\Delta\phi+\pi)} dy\ f_0^{(2)}(y) \qquad (3.3\text{-}5c)$$

$$B_{02} = \int_{2\pi-2\Delta\phi}^{\Delta\phi+\pi} dx\ f_0^{(1)}(x) \int_{4\pi-x}^{2(\Delta\phi+\pi)} dy\ f_2^{(2)}(y) \qquad (3.3\text{-}5d)$$

$$B_{11} = \int_{2\pi-2\Delta\phi}^{\Delta\phi+\pi} dx\ f_1^{(1)}(x) \int_{4\pi-x}^{2(\Delta\phi+\pi)} dy\ f_1^{(2)}(y) \qquad (3.3\text{-}5e)$$

and

$$B_{20} = \int_{2\pi-2\Delta\phi}^{\Delta\phi+\pi} dx\, f_2^{(1)}(x) \int_{4\pi-x}^{2(\Delta\phi+\pi)} dy\, f_0^{(2)}(y). \tag{3.3-5f}$$

Notice in (3.3-3) and (3.3-5) that the integration limits do not depend upon m; the significance of this fact is that for no jamming, when

$$f_m^{(1)}(x) = p_{0,m}^{(1)}\, p_\psi^{(1)}(x|\rho_N,\underline{\beta}) \tag{3.3-6a}$$

and

$$f_n^{(L-1)}(y) = p_{0,n-m}^{(L-1)}\, p_\psi^{(L-1)}(y|\rho_N,\underline{\beta}), \tag{3.3-6b}$$

then the click probabilities factor out, leaving identical integrals. That is, for this case

$$B_{m,n-m}^{(L)} = p_{0,m}^{(1)}\, p_{0,n-m}^{(L-1)}\, C_n, \tag{3.3-7a}$$

where

$$C_n = \int_{\max[\Delta\phi-\pi,2\pi n-(L-1)(\Delta\phi+\pi)]}^{\Delta\phi+\pi} dx\, p_\psi^{(1)}(x|\rho_N,\underline{\beta}) \int_{\max[(L-1)(\Delta\phi-\pi),2\pi n-x]}^{(L-1)(\Delta\phi+\pi)} dy\, p_\psi^{(L-1)}(y|\rho_N,\underline{\beta});$$

numerically, the integrals have to be computed once, rather than n + 1 times. However, for the general case, the click probabilities do not easily factor out of $f_m^{(1)}$ and $f_{n-m}^{(L-1)}$ [see (3.2-4), for example], and the double integral must be computed for each m. In view of the increasingly complex and time-consuming computations required by the direct method as the order of diversity L increases, we have developed numerical methods for implementing a characteristic function or transform approach.

### 3.3.2 Methodology Using Transform Approach

As we have noted, it is sufficient for BER calculations to model the single-sample pdf as having a finite maximum number of clicks:

$$p^{(1)}(x|\underline{\beta}) = \sum_{n=0}^{nmax} f_n(x+2\pi n;\underline{\beta}). \qquad (3.3\text{-}8)$$

This model is sufficient because numbers of clicks greater than nmax simply do not necessarily enter into the BER calculation. Given the pdf (3.3-8), the range of the differential phase (x) which is significant then is

$$\Delta\phi-\pi-2\pi nmax < x < \Delta\phi+\pi, \qquad (3.3\text{-}9)$$

Since the pdf is zero outside this range, we can consider numerical evaluation of the characteristic function, by means of the DFT (discrete Fourier transform) of the pdf. Theoretically, it is well understood that if

$$\varphi(\nu) = \int_{-\infty}^{\infty} dx\; e^{-j2\pi\nu x}\, p^{(1)}(x) = \mathcal{F}\left\{p^{(1)}(x)\right\} \qquad (3.3\text{-}10)$$

is the characteristic function for one sample, then

$$\left[\varphi(\nu)\right]^{L} = \mathcal{F}\left\{p^{(L)}(x)\right\}, \qquad (3.3\text{-}11)$$

that is, the $L$th power of $\varphi$ is the characteristic function for the sum of L independent and identically-distributed samples. The challenge is to utilize DFT methods and parameters which will provide sufficient accuracy to evaluate the BER.

### 3.3.2.1 DFT Size Considerations

Figure 3.3-1 illustrates the fact that, since the L = 1 pdf for a given pattern is nonzero for the range $\Delta\phi-\pi-2\pi nmax < x < \Delta\phi+\pi$, the L-hop pdf is nonzero for the range $L(\Delta\phi-\pi-2\ nmax) < x < L(\Delta\phi+\pi)$. Therefore, to avoid aliasing, the N-point DFT must consider pdf samples on an interval at least as long as

$$X = N\Delta x \geq L(nmax+1)2\pi \text{ radians.} \tag{3.3-12}$$

From (2.2-22b), a conservative estimate of nmax, the number of "significant" clicks, is nmax = L-1. Thus the interval should be at least $L^2 2\pi$ radians in length. At the same time, it is convenient to stipulate that in $\pi$ radians there are exactly an integer number of sampling intervals, that is,

$$\frac{\pi}{\Delta x} = \text{integer} \stackrel{\Delta}{=} n_0. \tag{3.3-13}$$

This implies that

$$X = N\Delta x = N \cdot \pi/n_0 \geq 2\pi L^2; \tag{3.3-14a}$$

or that the DFT size N is bounded by

$$N \geq 2\pi L^2/\Delta x = 2L^2 n_0 . \tag{3.3-14b}$$

For example, if we specify that $n_0 = 128$ samples are desired over $\pi$ radians ($\Delta x = \pi/128$), then the following DFT sizes are required:

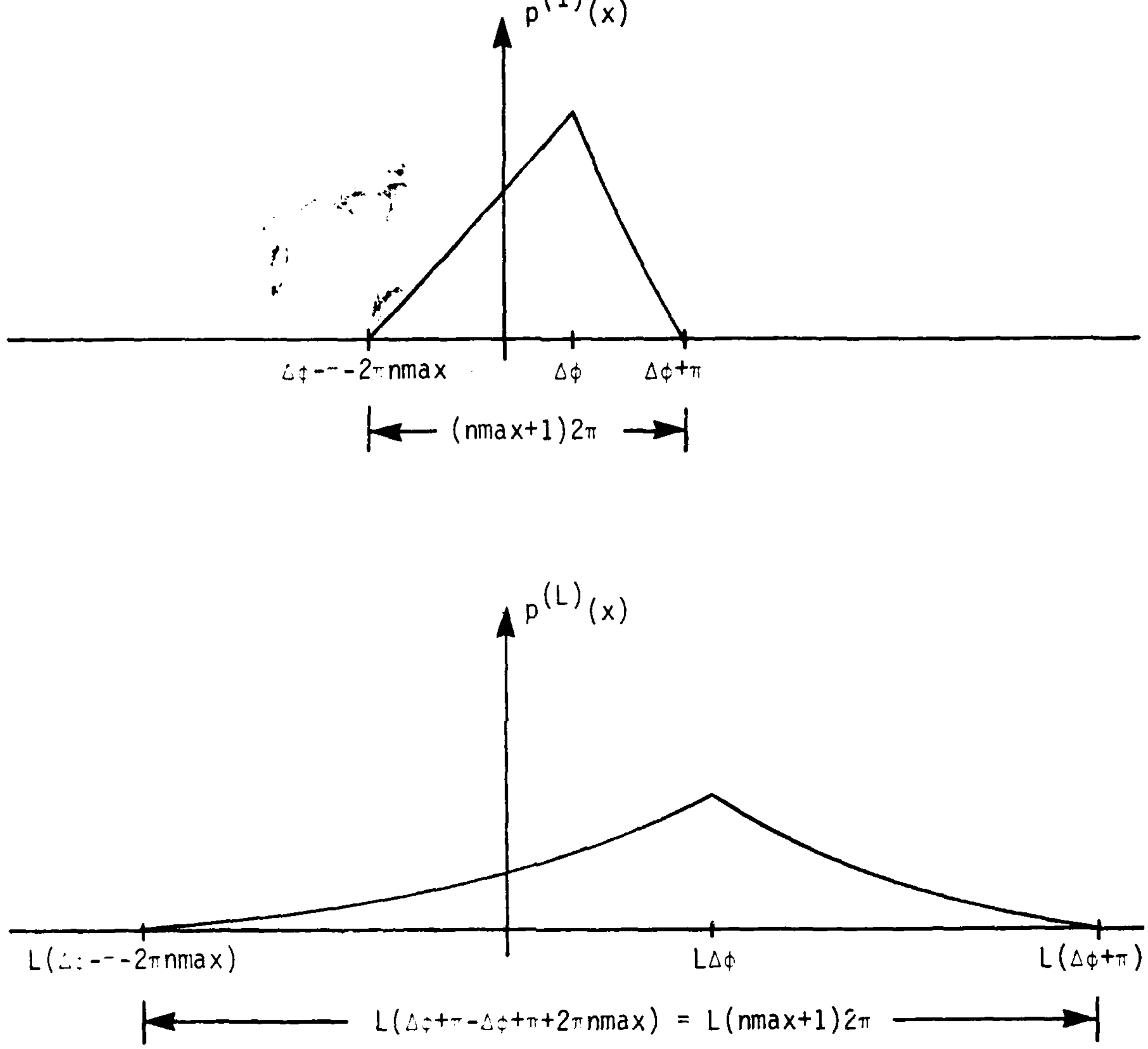



FIGURE 3.3-1 RANGE OF DIFFERENTIAL PHASES WITH NONZERO PDF FOR ONE AND FOR L HOPS/BIT

| L | minimum N | preferred N |
|---|---|---|
| 1 | 256 | 256 |
| 2 | 1024 | 2048 |
| 3 | 2304 | 4096 |
| 4 | 4096 | 8192 |

In this table, we choose a "preferred N" value that is the next higher power of two, in order to provide "zero fill" for interpolating the characteristic function with more closely-spaced points in the transform domain. It seems unlikely, however, that there is any "best" window or window length in this application of the DFT, since the effective width (standard deviation) of the pdf will vary with SNR, and it is impractical to try to match the two widths in some sense for each SNR value.

3.3.2.2 Formulation of BER Using DFT

The strategy we have adopted for calculating the BER is designed to exploit the fact that a finite number of clicks is involved in calculation of the probability of a correct decision, P(C) = 1-P(e). Thus our approach is to utilize DFT methods to calculate

$$P(C) = \frac{1}{4}\left[P(C|111) + 2P(C|011) + P(C|010)\right], \qquad (3.3\text{-}15)$$

where the conditional probability of a correct decision is

$$\begin{aligned} P(C|\underline{\beta}) &= \int_0^{L(\Delta\phi+\pi)} dx\, p^{(L)}(x|\underline{\beta}) \\ &= \int_0^{L(\Delta\phi+\pi)} dx \int_{-\infty}^{\infty} d\nu\, e^{j2\pi\nu x}\left[\int_{\Delta\phi-\pi-2\pi n_{max}}^{\Delta\phi+\pi} dy\, e^{-j2\pi\nu y} p^{(1)}(y|\underline{\beta})\right]^L . \qquad (3.3\text{-}16) \end{aligned}$$

We use Simpson's rule to perform the outer integral, with the L-diversity pdf approximated by

$$p^{(L)}(m\Delta x) \approx \Delta\nu \sum_{r=0}^{N-1} e^{j2\pi mr\Delta x\Delta\nu}\left[\Delta y \sum_{n=0}^{N-1} e^{-j2\pi rn\Delta y\Delta\nu}\hat{p}(n\Delta y)\right]^L , \qquad (3.3\text{-}17)$$

in which $\hat{p}$ is the periodic extension of $p^{(1)}$. Since $x = \Delta y = X/N$ and $\Delta\nu = 1/X$, then $\Delta x \cdot \Delta y = 1/N$ and (3.3-17) becomes

$$p^{(L)}(m\Delta x) \approx X^{L-1}\ \mathrm{IDFT}\left\{[\mathrm{DFT}(\hat{p})]^L\right\}. \qquad (3.3\text{-}18)$$

The use of the inverse DFT (IDFT) in (3.3-17) and (3.3-18) is based on the fact that the finite Fourier transform of the periodic extension of the (finite domain) L = 1 pdf is (a) equal to the pdf's characteristic function at the points $\nu = n \cdot \Delta\nu$, and (b) periodic with period $1/\Delta x$ when the finite Fourier transform is approximated by a DFT. That is,

$$\hat{\varphi}^{(1)}(\nu) = \mathcal{F}_X\{\hat{p}(x)\}$$

$$= \int_0^X dx\, e^{-j2\pi\nu x} \sum_{k=-\infty}^{\infty} p^{(1)}(x+kX)$$

$$= \int_0^X dx\, e^{-j2\pi\nu x}\, p^{(1)}(x) + e^{j2\pi\nu X}\int_{-X}^{0} dx\, e^{-j2\pi\nu x} p^{(1)}(x)$$

$$= \varphi^{(1)}(\nu),\ \nu = n \cdot \Delta\nu \equiv n/X; \qquad (3.3\text{-}19)$$

and if

$$\hat{\varphi}_N^{(1)}(\nu) \overset{\Delta}{=} \mathrm{DFT}\{\hat{p}(x)\}, \tag{3.3-20a}$$

then

$$\hat{\varphi}_N^{(1)}[(n+N)\Delta\nu] = \hat{\varphi}_N^{(1)}(n\Delta\nu). \tag{3.3-20b}$$

Therefore approximating the inverse Fourier transform integral by a summation from $\nu = -N\Delta\nu/2$ to $\nu = N\Delta\nu/2$ is the same as a summation from $\nu = 0$ to $\nu = (N-1)\Delta\nu$.

The approximation, besides utilizing a discrete sum to calculate an integral, is predicated on the characteristic function's vanishing for $|\nu| \rightarrow N\Delta\nu/2$; otherwise, it will be distorted due to aliasing. This vanishing never exactly occurs, since the pdf is nonzero over a finite interval. However, the more smoothly the pdf decreases at its end points, the less "bandwidth" is required, and aliasing of the characteristic function is minimized. For example, the least "smooth" pdf occurs for zero SNR, and is given by

$$p^{(1)}(x;\rho=0) = \frac{1}{2\pi}\sum_{n=0}^{nmax} p_n[u(\Delta\phi-\pi+2n\pi) - u(\Delta\phi+\pi+2n\pi)], \tag{3.3-21}$$

in which $u(\cdot)$ is the unit step function and the $\{p_n\}$ are click probabilities. The magnitude of the corresponding characteristic function is bounded by

$$|\varphi^{(1)}(\nu)| \leq \frac{1}{2\pi^2|\nu|} = \frac{1}{128\pi} \text{ for } \nu = N\Delta\nu/2 \text{ and } \Delta x = \pi/128. \qquad (3.3\text{-}22)$$

This is 26 dB down from the maximum value of $\varphi(0) = 1$. A triangular pdf for no clicks would be 52 dB down at the N/2 folding point, and a Gaussian density with $\sigma^2 = 1/\varepsilon$ would be down the following amounts for different SNR values, $\varepsilon$, and $\Delta x = \pi/128$:

| $\underline{\varepsilon}$ | Attenuation Bound at $N\Delta\nu/2$ |
|---|---|
| (expression) | $0.54/\varepsilon(\Delta x)^2$ (dB) |
| 0 dB | 901 dB |
| 10 dB | 90 dB |
| 20 dB | 9 dB. |

3.3.2.3 Normalizations

Referring now to equations (3.3-16) and (3.3-17), the characteristic function for L = 1 is approximated by

$$\varphi^{(1)}(\nu) = \int_{\Delta\varepsilon-\pi-2\pi n\max}^{\Delta\varepsilon+\pi} dx\, e^{-j2\pi\nu x}\, p^{(1)}(x) \approx X \;\; \mathrm{DFT}\{\acute{p}\}. \qquad (3.3.23)$$

As a means for controlling the approximation error, we utilize the fact that

$$\varphi^{(1)}(0) = \sum_{n=0}^{nmax} [(1-\gamma)p_{0,n} + \gamma p_{1,n}]. \quad (3.3\text{-}24)$$

Therefore for each intersymbol interference pattern we adjust the DFT values by a normalization factor to produce

$$DFT(k)' = \frac{DFT(k)}{NF}, \quad (3.3\text{-}25a)$$

where

$$NF = X \cdot DFT(0)/\varphi^{(1)}(0). \quad (3.3\text{-}25b)$$

### 3.3.2.4 Averaging Over Data Patterns

To minimize the number of inverse DFT operations, we perform averaging over data patterns prior to the inverse DFT. That is, the pdf for the diversity sum is computed as

$$p^{(L)}(max) = X^{L-1}\ \mathrm{IDFT}\left\{ \frac{1}{4}[DFT(\hat{p},111)/NF(111)]^L + \frac{1}{2}[DFT(\hat{p},011)/NF(011)]^L + \frac{1}{4}[DFT(\hat{p},010)/NF(010)]^L \right\}. \quad (3.3\text{-}26)$$

### 3.3.3 Results Using DFT Method

Since we previously have calculated the FH/CPFSK bit error probability for L = 2 by the direct method, the accuracy of the DFT method's results may be discerned by making comparisons for this case. Also, the values of the normalization factors for the different patterns give a general indication of how well the discretized pdf is representing the actual, continuous distribution.

The worst-case BER was found by computing a few points on BER vs $\gamma$ curves like the ones shown in Figure 3.2-2, just enough points to establish the P(e) maximum with respect to $\gamma$. The program included in Appendix D was used, and each point required between five and ten minutes of computation, depending on L. Normalization factors typically were between 0.95 and 1.05, indicating that numerical integration of the L = 1 pdf using $2\pi/\Delta x = 256$ points and a rectangular rule would yield about a 5% error, if the factors were not employed.

### 3.3.3.1 Results for $E_b/N_0 = 15$ dB

Figure 3.3-2 summarizes the worst-case partial-band noise jamming performance of FH/CPFSK using discriminator detection and linear diversity combining. The curves were drawn using the data in Table 3.3-1. We have already noted in Section 3.2 that the L = 2 performance is about 4.4 dB worst than for L = 1 (no diversity); these previous results of direct calculation are included in the figure for reference. Worst-case BER results for L = 2 using the DFT method are also presented for the portion of the graph lying between 10 and 30 dB. Pictorially, the L = 2 results using the two different methods are barely distinguishable; the data in Table 3.3-1 indicates that the DFT method gives a BER about 3% high, relative to the direct method. This was considered to be an acceptable degree of agreement between the two methods, so that further development of the computer program, such as more elaborate normalizations, was not pursued.

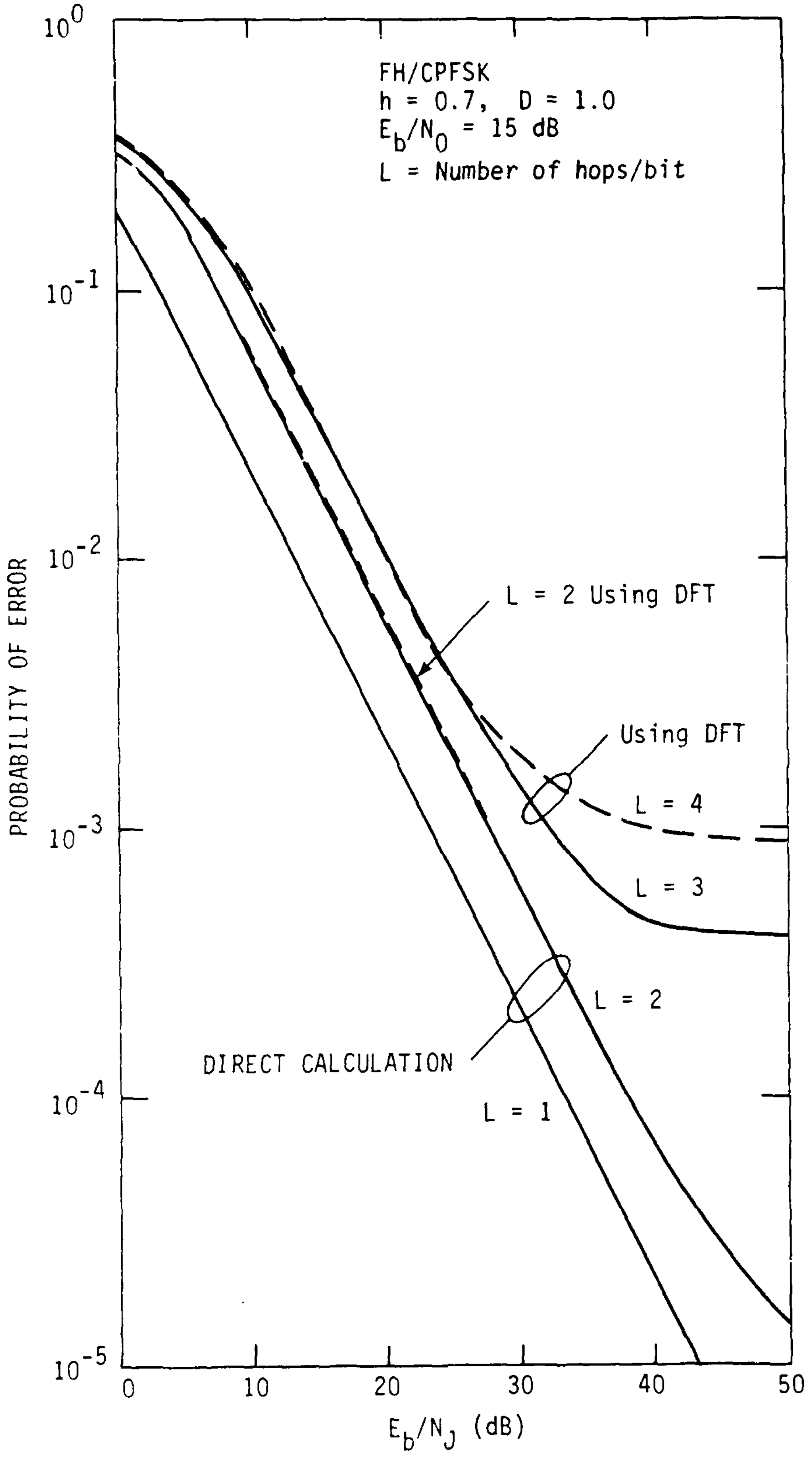



FIGURE 3.3-2 WORST-CASE FH/CPFSK BER VS $E_b/N_J$ FOR $E_b/N_0$ = 15 dB AND THE NUMBER OF HOPS/BIT (L) VARIED

TABLE 3.3-1 DATA FOR FIGURE 3.3-2

| $E_b/N_J$ (dB) | Direct Calculation | | Calculation using DFT method | | |
|---|---|---|---|---|---|
| | $L = 1$ | $L = 2$ | $L = 2$ | $L = 3$ | $L = 4$ |
| 0 | 0.196 | 0.322 | | 0.37 | 0.386 |
| 5 | 0.067 | 0.182 | | | |
| 10 | 0.021 | 0.058 | 0.0597 | 0.097 | 0.108 |
| 15 | 6.7(-3) | 1.83(-2) | | 3.15(-2) | 3.15(-2) |
| 20 | 2.1(-3) | 5.8(-3) | 5.99(-3) | 1.03(-2) | 1.02(-2) |
| 25 | 6.7(-4) | 1.84(-3) | | 3.52(-3) | 3.81(-3) |
| 30 | 2.1(-4) | 5.9(-4) | 6.07(-4) | 1.44(-3) | 1.80(-3) |
| 40 | 2.1(-5) | 6.7(-5) | | 4.45(-4) | 9.7(-4) |
| 50 | 2.1(-6) | 1.4(-5) | | 3.9(-4) | 8.9(-4) |

The L = 3 results in Figure 3.3-2 reflect a 2.4 dB worse performance than for L = 2, when the jammer power is strong (low $E_b/N_J$). As $E_b/N_J$ increases, the BER for this number of hops/bit converges to an asymptote of about $4 \times 10^{-4}$, the performance for L = 3 and no jamming. Evidently the unjammed noncoherent combining loss at $E_b/N_0$ = 15 dB is about 5.2 dB, because (from Figure 3.1-3) a $4 \times 10^{-4}$ BER for L = 1 occurs for $E_b/N_0 \simeq 9.8$ dB. This loss contrasts with the amount in strong jamming, about 6.8 dB.

As anticipated, the L = 4 results in Figure 3.3-2 follow the trend of the BER increasing with L, at least for strong jamming. We observe that for 15 dB < $E_b/N_J$ < 25 dB, the L = 4 worst-case performance dips below that of L = 3, before settling to its unjammed value of $9 \times 10^{-4}$ (5.7 dB unjammed noncoherent combining loss). This phenomenon is very interesting, since we are seeking a BER behavior which decreases with L. But in this figure, we cannot discern the trend because the L = 3 and L = 4 performances are influenced so much by thermal noise.

### 3.3 Results for $E_b/N_0$ = 20 dB

In Figure 3.3-3 we show L = 2, 3, and 4 BER results for $E_b/N_0$ = 20 dB, plotted from the data in Table 3.3-2. For each diversity value (L), the thermal noise is not influential - that is, the unjammed BER is much less than $10^{-5}$ - so we can observe the trend of the relative performances for L = 3 and L = 4. What we see is the fact that, for negligible thermal noise, the worst-case BER performance for L = 4 lies between, and parallel to, those for L = 2 and L = 3. Thus, although a slight improvement is made (about 1 dB), there is no "diversity gain" improvement for higher L. If there were such an improvement, the negative slope of the (log) P(e) curve vs $E_b/N_J$ in dB would

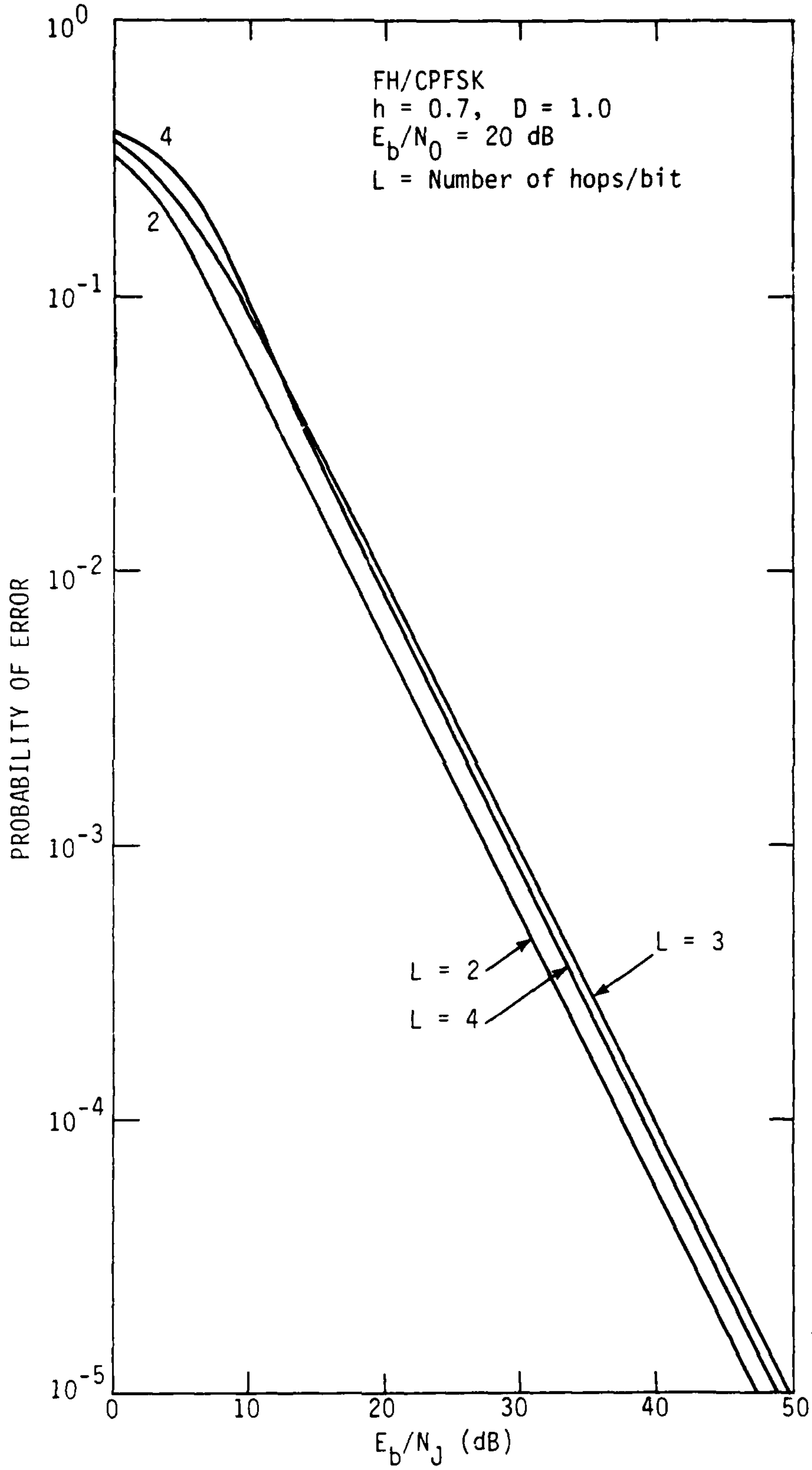



FIGURE 3.3-3 WORST-CASE FH/CPFSK BER VS $E_b/N_J$ FOR $E_b/N_0$ = 20 dB AND THE NUMBER OF HOPS/BIT (L) VARIED

TABLE 3.3-2 DATA FOR FIGURE 3.3-3

| $E_b/N_J$ (dB) | L = 2 | | L = 3 | | L = 4 | |
|---|---|---|---|---|---|---|
| | $P_{WC}$ | $\gamma_{WC}$ | $P_{WC}$ | $\gamma_{WC}$ | $P_{WC}$ | $\gamma_{WC}$ |
| 0 | 0.329 | 1.0 | 0.37 | 1.0 | 0.385 | 1.0 |
| 5 | 0.173 | 1.0 | 0.218 | 1.0 | 0.263 | 1.0 |
| 7.5 | | | | | 0.17 | 1.0 |
| 10 | 5.5(2) | 0.3 | 8.6(-2) | 0.48 | 8.9(-2) | 0.55 |
| 15 | 1.74(-2) | 9.5(-2) | 2.8(-2) | 0.15 | 2.6(-2) | 0.16 |
| 20 | 5.5(-3) | 3.0(-2) | 9.0(-3) | 4.8(-2) | 7.8(-3) | 5.0(-2) |
| 25 | | | 2.8(-3) | 1.5(-2) | | |
| 30 | 5.6(-4) | 3.0(-3) | 9.0(-4) | 4.8(-3) | 7.8(-4) | 5.0(-3) |
| 40 | 5.5(-5) | 3.0(-4) | 9.0(-5) | 4.8(-4) | 7.7(-5) | 5.0(-4) |
| 50 | 5.5(-6) | 3.0(-5) | 9.0(-6) | 4.8(-5) | 7.7(-6) | 5.0(-5) |

Parameters: $E_b/N_0$ = 20 dB, h = 0.7, D = 1.0

be greater than unity.

3.3.3.3 Explanation of the Observed Diversity Behavior

What then accounts for the slight worst-case BER improvement going from $L = 3$ to $L = 4$? The apparent explanation can be made with the help of Figures 3.3-4 to 3.3-7. In Figure 3.3-4, we show the pattern-averaged pdf's for a single hop sample and for the sum of two hop samples when $L = 2$ for particular values of $\gamma$, $E_b/N_0$, and $E_b/N_J$. The linear scale is such that the click contributions are not noticeable, so in Figure 3.3-5 we repeat the same information, using a logarithmic scale. The superposition of the three pattern-dependent pdf's constituting the averaged pdf for $L = 1$ is now evident in Figure 3.3-5, and we call the reader's attention to two properties of the two-sample pdf: (1) its major peak is shifted to the right, located at $2\Delta\phi$, compared to $\Delta\phi$ for one sample; (2) the width of its lobes are increased over that for one sample. Thus, while for $L = 2$ the pdf shifts to the right, tending to decrease the error, it also is spreading out, and furthermore the "mass" under the one-click lobe is greater than for one sample; both these latter trends tend to increase the error.

In Figures 3.3-6 and 3.3-7 we present similar pdf illustrations, for $L = 3$ and $L = 4$, respectively. The trend is for the nonzero click lobes to become more significant as L increases, since the SNR is decreasing for constant bit energy ($\epsilon_N = (E_b/N_0)/L$). However, as L goes from 3 to 4, note that the peak of the 1-click lobe crosses zero. This means that for high SNR and $L = 4$, a correct decision is made even when there is a single click, whereas for $L = 3$ a correct decision is not made if there are any clicks at all.

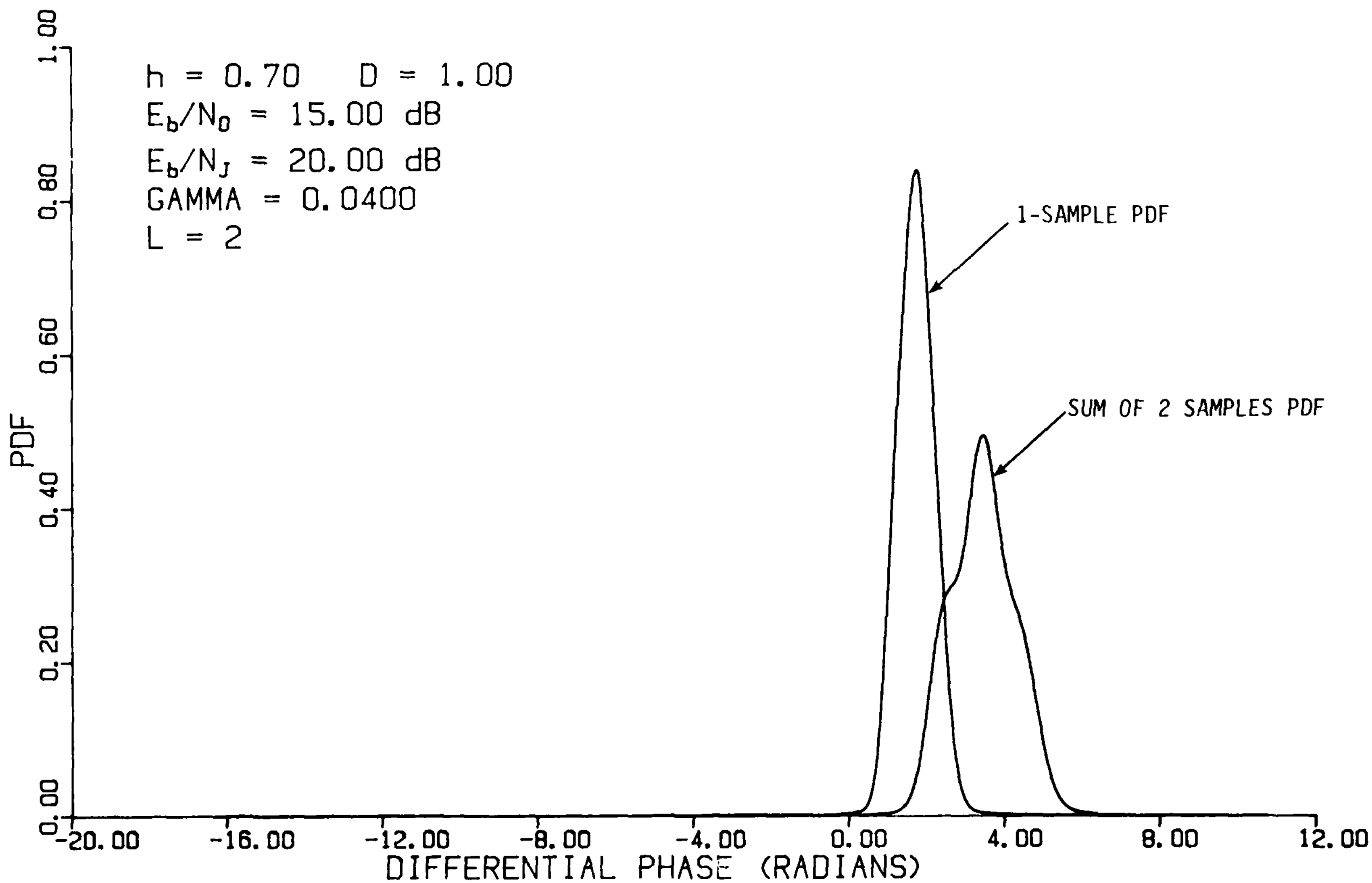



FIGURE 3.3-4 PATTERN-AVERAGED DIFFERENTIAL PHASE SUM PDF'S FOR L = 2 (LINEAR SCALE)

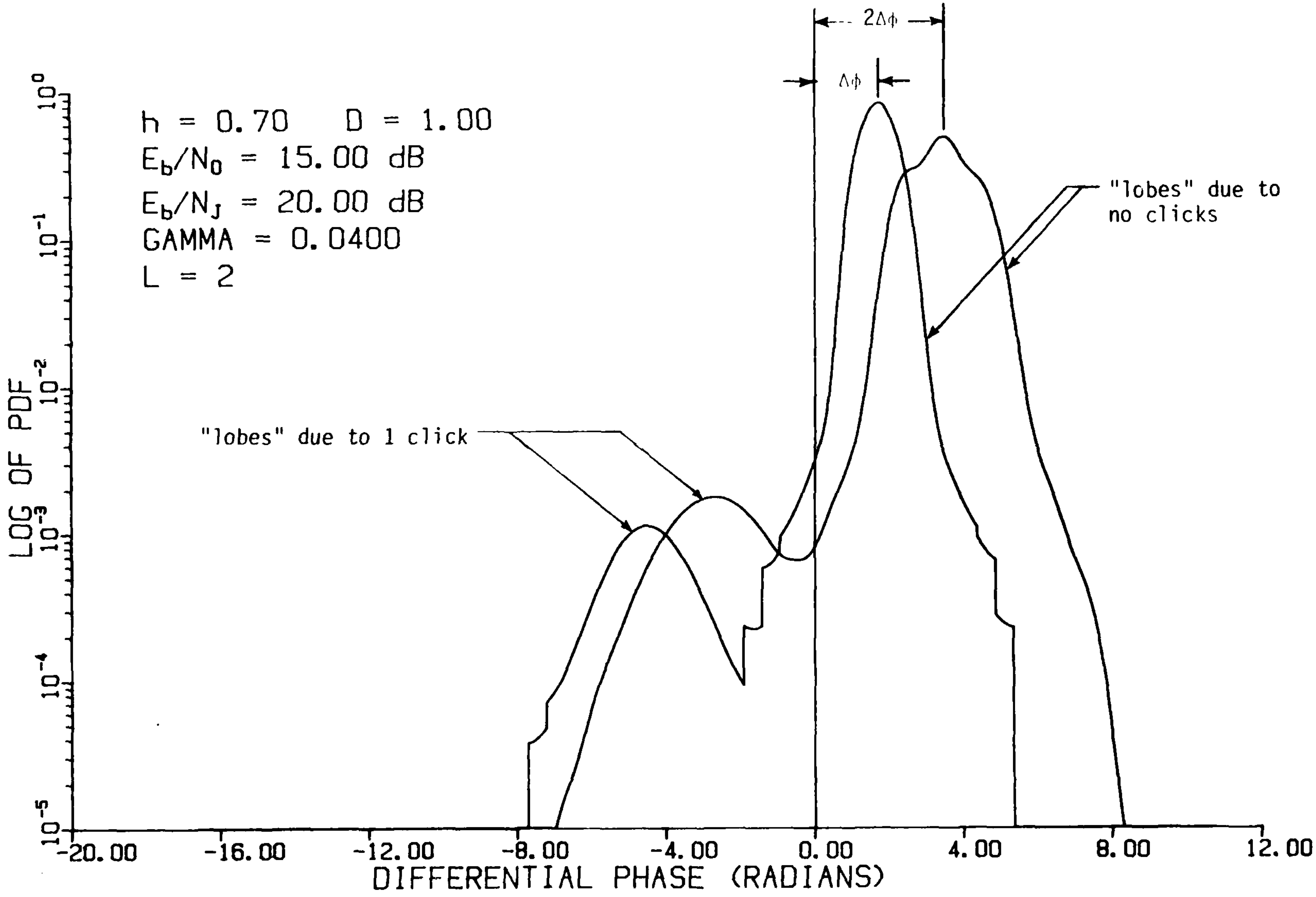



FIGURE 3.3-5 PATTERN-AVERAGED DIFFERENTIAL PHASE SUM PDF'S FOR L = 2 (LOG SCALE)

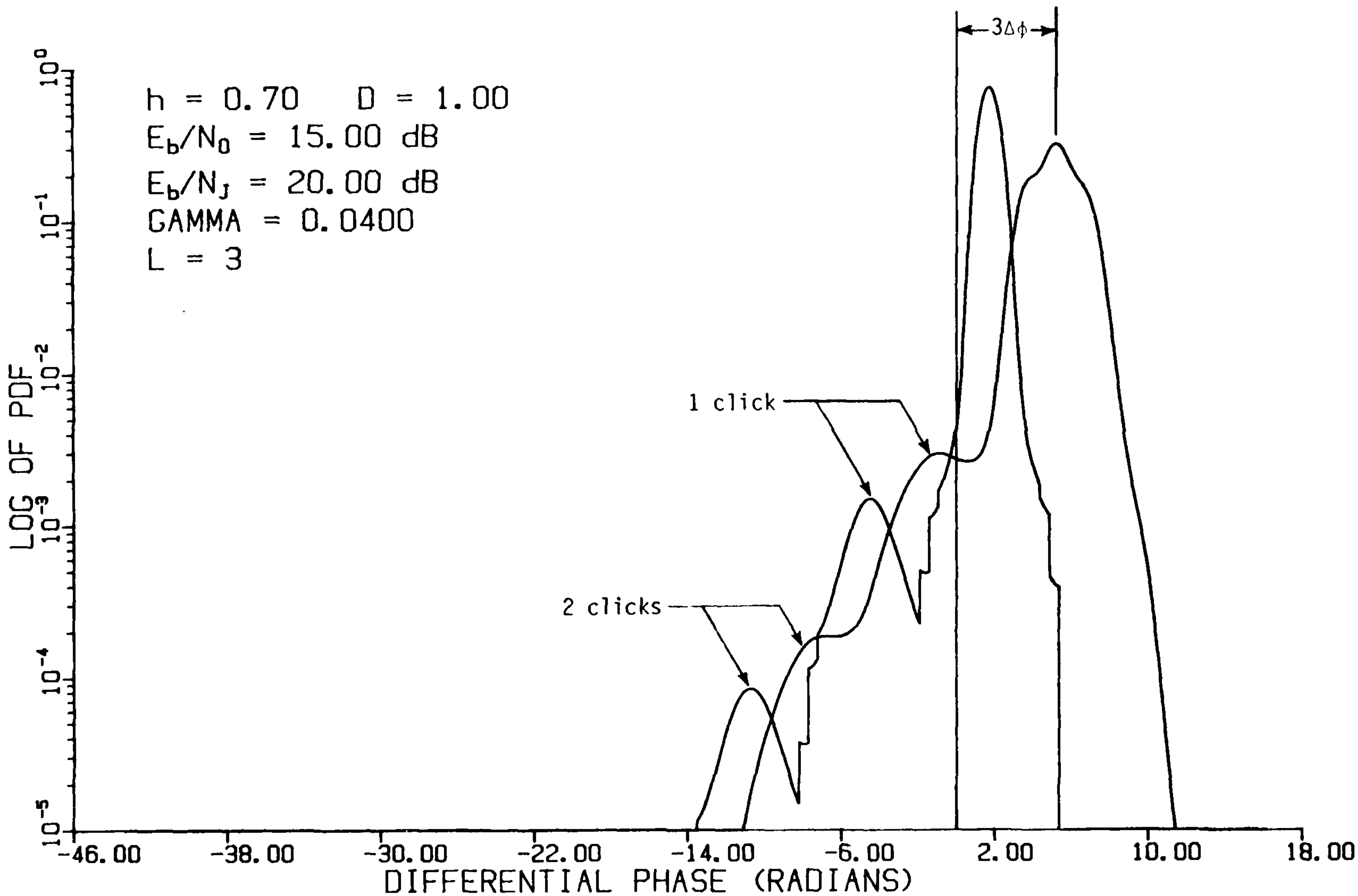



FIGURE 3.3-6 PATTERN-AVERAGED DIFFERENTIAL PHASE SUM PDF'S FOR L = 3 (LOG SCALE)

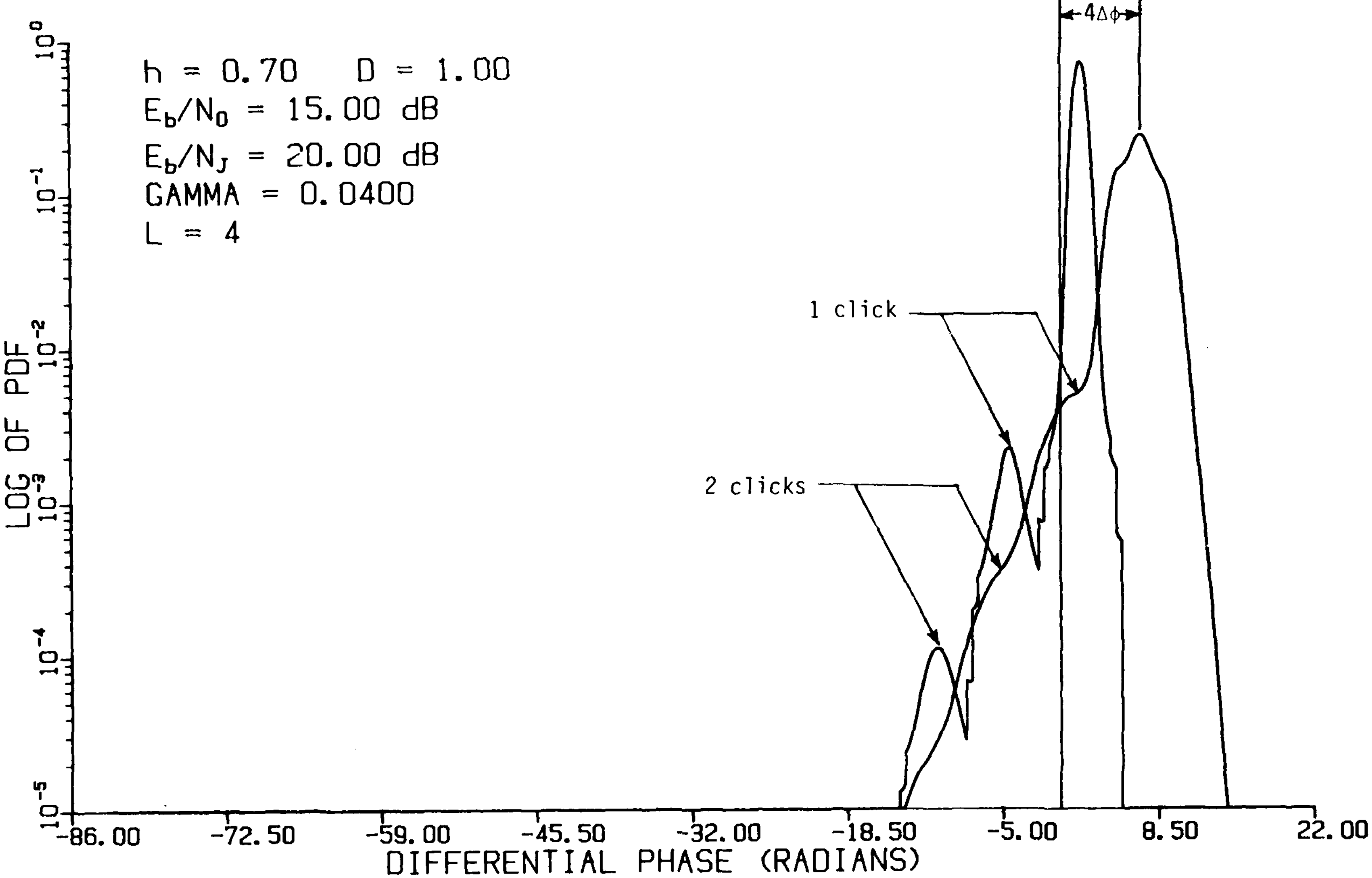



FIGURE 3.3-7 PATTERN-AVERAGED DIFFERENTIAL PHASE SUM PDF'S FOR L = 4 (LOG SCALE)

Thus L = 4 has a slight advantage over L = 3, but does not offer a "diversity gain" improvement.

### 3.3.3.4 Significance of the Diversity Behavior

The BER results shown in Figures 3.3-2 and 3.3-3 demonstrate the fact that a diversity gain against worst-case partial-band noise jamming is not realized for the linear combining of FH/CPFSK differential phase samples from different hops. The same fact is true for FH/BFSK, but it was plausible to conjecture that the nonlinear demodulation procedures for CPFSK (e.g., the bandpass limiter) would provide the sort of nonlinear processing that produces a diversity gain. Now that this conjecture has been disproved, we observe with hindsight that the limiter (ideally) has no effect on the phase of the signal, and that there is no mechanism in the limiter/discriminator receiver which acts to de-emphasize samples from jammed hops.

It is easy to show that some form of diversity gain is possible with the appropriate processing, at least for low thermal noise. Consider, for example, the scheme in which it is assumed that perfect side information is available on which hops are jammed, and samples from jammed hops are excluded from the diversity sum unless all hops are jammed. Then, for no thermal noise, no error occurs unless all hops are jammed:

$$P(e) = \gamma^{L} g_L(\gamma R_J), \quad R_J \equiv E_b/N_J, \tag{3.3-27}$$

where $g_L(\cdot)$ is the BER vs. $E_b/N_0$ function for the sum of L samples. Differentiation with respect to $\gamma$ yields the equation

$$L\ g_L(x) + x\ g_L'(x) = 0, \quad x \equiv \gamma R_J. \tag{3.3-28}$$

Clearly, if this equation is solved for $x = x_L$, then the worst-case partial-band jamming fraction is

$$\gamma_{WC} = \begin{cases} \dfrac{x_L}{R_J}, & R_J > x_L; \\ \\ 1, & R_J < x_L. \end{cases} \tag{3.3-29}$$

Substituting the value of $\gamma_{WC}$ back into (3.3-27), we find that

$$P_{WC} = \begin{cases} \left(\dfrac{x_L}{R_J}\right)^L g_L(x_L), & R_J > x_L; \\ \\ g_L(R_J), & R_J < x_L. \end{cases} \tag{3.3-31}$$

That is, $P_{WC}$ is proportional to $R_J^{-L}$ for $R_J > x_L$. On a log P(e) vs $E_b/N_J$ (dB) plot, $P_{WC}$ is a straight line with slope -L, tangent to the $\gamma = 1$ error curve at $R_J = x_L$.

Using data from Tables 3.3-1 and 3.3-2 for $\gamma = 1$, and constructing tangents with slopes equal to -L, we obtain the predicted ideal diversity performances for FH/CPFSK in worst-case PBNJ as shown in Figure 3.3-8. The higher L curves must eventually cross those for lower L because of their greater slope, and indeed do as shown. The figure illustrates that a diversity gain is realized for this ideal situation, and if the optimum value of L is always used, the worst-case BER can be made to approach within 5 or 6 dB of the unjammed CPFSK performance, compared to over 30 dB without diversity.

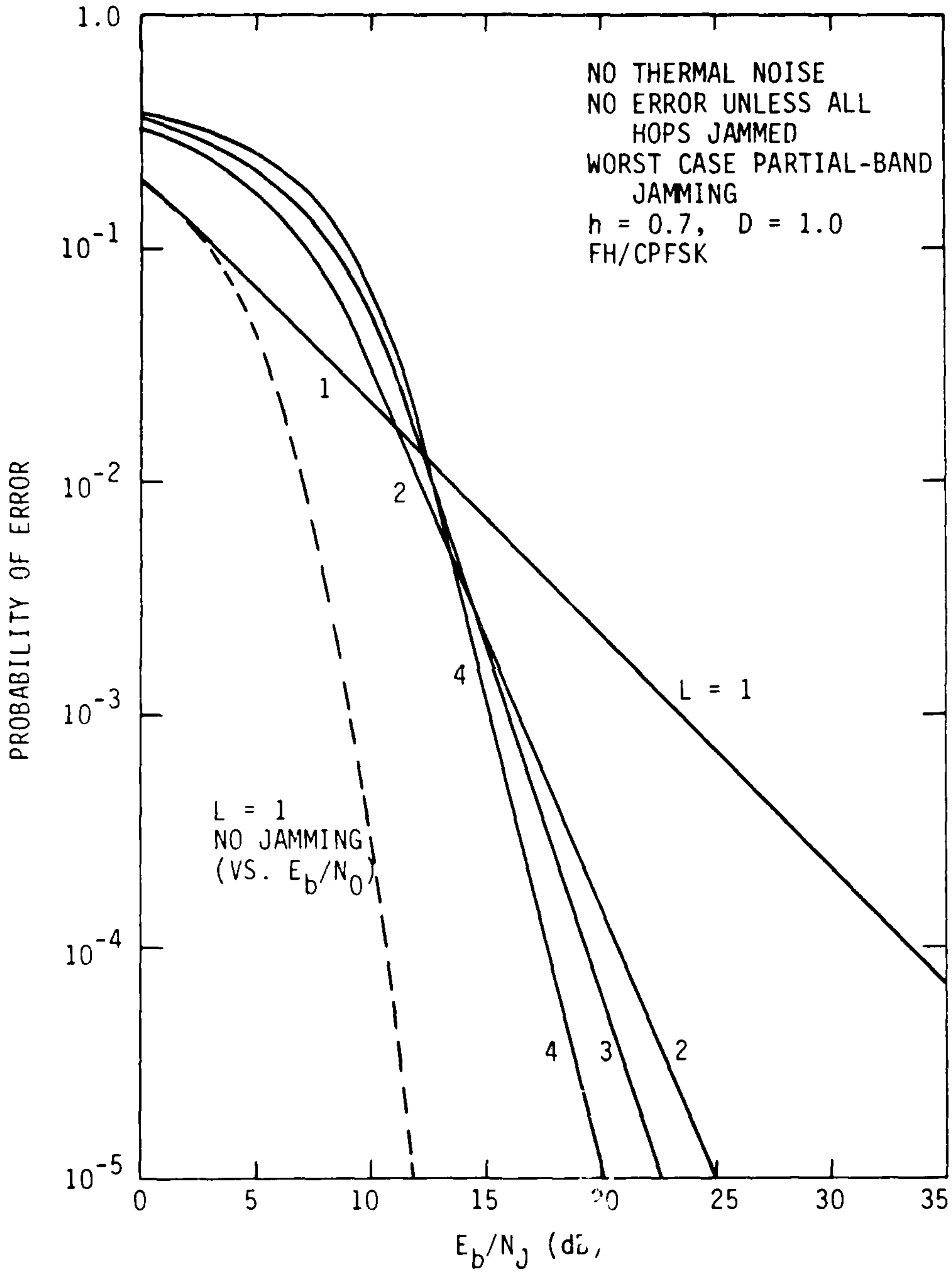



FIGURE 3.3-8 FH/CPFSK DIVERSITY SUM PERFORMANCE ASSUMING PERFECT SIDE INFORMATION AND NO THERMAL NOISE

In practice, thermal noise is always present, giving rise to noncoherent combining losses which prevent achievement of the ideal diversity gains pictured in Figure 3.3-8. However, these results do suggest that some form of combining the differential phase samples from the different hops will accomplish a diversity improvement of some degree.

## 4.0 EVALUATION OF DIVERSITY PERFORMANCE USING DIFFERENTIAL DETECTION

An alternate receiver configuration for FH/CPFSK is one in which the limiter-discriminator is replaced with a differential detector. In this section, we find the BER obtained by FH/CPFSK in partial-band noise jamming when a differential detector is used.

### 4.1 ANALYSIS FOR NO DIVERSITY

The analysis of differential detection of narrowband FM presented by Simon and Wang [11] relies on the phase distribution theory of Pawula, Rice, and Roberts [8], applied to the binary FM communications problem by Pawula [3]. The derivation of the error probability using this theory is somewhat involved. In this section, we present a simple derivation of the binary FM bit error rate (BER) using differential detection.

The differential detector, shown in Figure 4.1-1, develops the output

$$z(t) = \frac{1}{2} R(t)R(t-T)\sin[\theta(t)-\theta(t-T)] \tag{4.1-1}$$

from the narrowband waveform

$$x(t) = R(t)\cos[\omega_0 t + \theta(t)] \tag{4.1-2a}$$

$$= x_c(t)\cos\omega_0 t - x_s(t)\sin\omega_0 t. \tag{4.1-2b}$$

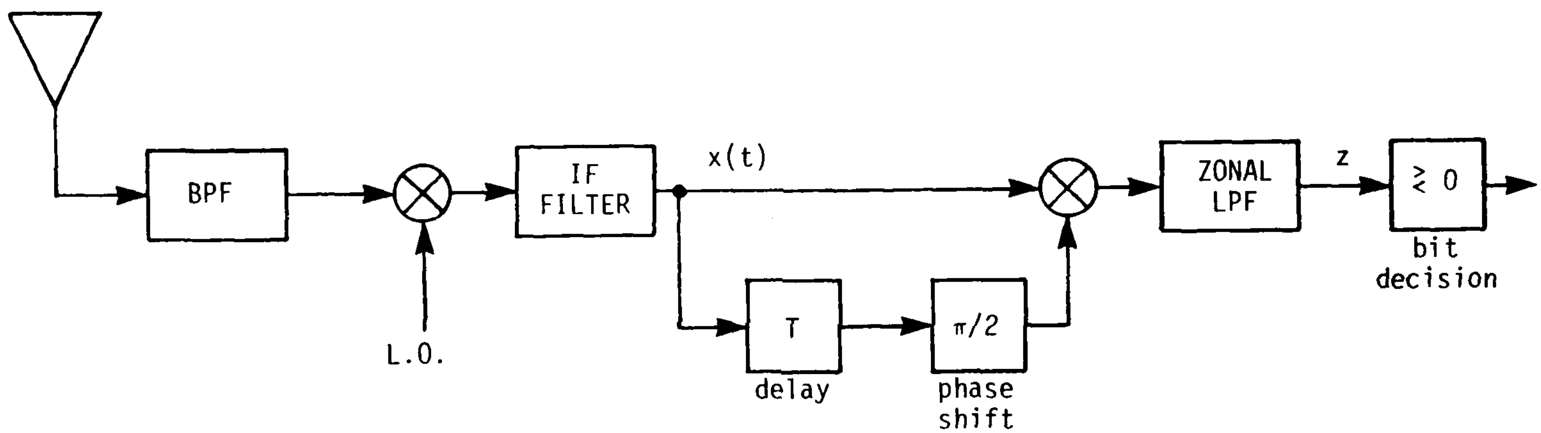



FIGURE 4.1-1 DIFFERENTIAL DETECTOR FOR NARROWBAND FM

The rationale for using this detector is that, for binary FM or CPFSK (continuous phase frequency-shift-keying), the bit information is contained in the sign of the difference between the signal phase values at bit sampling times $t = kT$. The sign of the quantity z at the sampling times therefore can be used for the bit decision. Without noise $x(t)$ is the signal

$$s(t) = a(t)\sqrt{2S}\,\cos[\omega_0 t + \phi(t)]\,, \tag{4.1-3}$$

where S is the signal power, $a(t)$ is amplitude modulation induced by the receiver filtering, and $\phi(t)$ is the information phase modulation waveform after distortion due to receiver filtering. Prior to this filtering, the signal has constant amplitude and the bit information is conveyed by the instantaneous frequency, which is either $f_0 + f_d$ for a "mark" or $f_0 - f_d$ for a "space". Therefore, neglecting distortion, the phase difference at $t_k = kT$ is

$$\Delta\phi = \phi(t_k) - \phi(t_k - T) = \pm\pi h, \tag{4.1-4}$$

where $h = 2f_dT$ is the modulation index.

Our approach to analyzing the BER for the differential detector recognizes that the random variable z is a quadratic form in the quadrature components $x_c(t)$ and $x_s(t)$:

$$z = \frac{1}{2}\,[x_s(t)x_c(t-T) - x_c(t)x_s(t-T)]. \tag{4.1-5}$$

With this understanding, as shown below we can apply a convenient equivalence for random variables of this type.

4.1.1 A Statistical Equivalence

Let the four-dimensional vector of random variables $\underline{x} = (x_1,x_2,x_3,x_4)^T$ be multivariate Gaussian, with mean vector $\underline{m}_x = (m_1,m_2,m_3,m_4)^T$ and covariance matrix

$$K_x(\eta,\xi) = \begin{bmatrix} \sigma_1^2 & 0 & \eta\sigma_1\sigma_2 & \xi\sigma_1\sigma_2 \\ 0 & \sigma_1^2 & -\xi\sigma_1\sigma_2 & \eta\sigma_1\sigma_2 \\ \eta\sigma_1\sigma_2 & -\xi\sigma_1\sigma_2 & \sigma_2^2 & 0 \\ \xi\sigma_1\sigma_2 & \eta\sigma_1\sigma_2 & 0 & \sigma_2^2 \end{bmatrix}. \tag{4.1-6}$$

We use $\underline{a}^T$ to denote the transpose of the column vector $\underline{a}$. In [12] and [13] it is shown that the quadratic form

$$y = \frac{1}{2}(x_1x_3 + x_2x_4) \tag{4.1-7}$$

is equal to the difference of two scaled, independent noncentral chi-squared random variables with two degrees of freedom, denoted by

$$y \sim c_1\chi^2(2;d_1) - c_2\chi^2(2;d_2), \tag{4.1-8}$$

where $d_1$ and $d_2$ are the noncentrality parameters. In terms of the components of $\underline{m}_x$ and $K_x$, the parameters are

$$c_{1,2} = \frac{1}{4}\sigma_1\sigma_2(\sqrt{1-\xi^2} \pm \eta) \tag{4.1-9a}$$

$$d_{1,2} = \frac{\sigma_2^2(m_1^2+m_2^2) + \sigma_1^2(m_3^2+m_4^2) \pm 2\sigma_1\sigma_2\sqrt{1-\xi^2}(m_1m_3+m_2m_4) - 2\sigma_1\sigma_2\xi(m_1m_4-m_2m_3)}{2\sigma_1^2\sigma_2^2\sqrt{1-\xi^2}\,(\sqrt{1-\xi^2} \pm \eta)}. \tag{4.1-9b}$$

### 4.1.2 Application of the Equivalence

Comparing (4.1-5) and (4.1-7), we define

$$\begin{bmatrix} x_c(t) \\ x_s(t) \\ -x_s(t-T) \\ x_c(t-T) \end{bmatrix} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \tag{4.1-10}$$

In this case, the mean vector $\underline{m}_x$ is

$$\begin{bmatrix} a(t)\sqrt{2S}\cos\phi(t) \\ a(t)\sqrt{2S}\sin\phi(t) \\ -a(t-T)\sqrt{2S}\sin\phi(t-T) \\ a(t-T)\sqrt{2S}\cos\phi(t-T) \end{bmatrix} = \begin{bmatrix} A_1\cos\phi_1 \\ A_1\sin\phi_1 \\ -A_2\sin\phi_2 \\ A_2\cos\phi_2 \end{bmatrix}. \tag{4.1-11}$$

For stationary bandpass Gaussian noise, $\sigma_1^2 = \sigma_2^2 = N_0W_N$, where $W_N = W_{IF}$ is the noise bandwidth of the receiver filter, and the autocorrelation function is

$$R_n(\tau) = N_0W_N[r(\tau)\cos\omega_0\tau + \lambda(\tau)\sin\omega_0\tau]; \tag{4.1-12}$$

thus $\xi = r(T) \equiv r$, and $\eta = -\lambda(T) \equiv -\lambda$. The cross-quadrature correlation coefficient $\lambda$ is zero if the filter passband is symmetric about the center frequency $f_0$.

Substituting appropriately in (4.1-8) and (4.1-9), then, we find that the differential detector output has the distribution

$$z \sim c_1 \chi^2(2;d_1) - c_2 \chi^2(2,d_2) \tag{4.1-13a}$$

with

$$c_{1,2} = \frac{1}{4} N_0 W_N \left(\sqrt{1-r^2} \mp \lambda\right) \tag{4.1-13b}$$

and

$$d_{1,2} = \frac{\rho_1 + \rho_2 \pm 2\sqrt{\rho_1\rho_2}\ \sqrt{1-r^2}\sin\Delta\phi - 2\sqrt{\rho_1\rho_2}\, r\cos\Delta\phi}{\sqrt{1-r^2}\ (\sqrt{1-r^2} \mp \lambda)} . \tag{4.1-13c}$$

In (4.1-13c) we define SNR's $\rho_i \triangleq A_i^2/2N_0W_N$ for $i = 1,2$. Further, defining $U \triangleq (\rho_1+\rho_2)/2$ and $W \triangleq \sqrt{\rho_1\rho_2}$, we have

$$d_{1,2} = \frac{2\{U - r\, W\cos\Delta\phi \pm \sqrt{1-r^2}\, W \sin\Delta\phi\}}{\sqrt{1-r^2}\ (\sqrt{1-r^2} \mp \lambda)} . \tag{4.1-14}$$

### 4.1.3 BER for L=1 Without Jamming

Now since the quantities U, W, and $\Delta\phi$ are affected by intersymbol interference, the BER must be averaged over possible bit patterns. Using the symmetry of complementary patterns, the BER can be written

$$
\begin{aligned}
P(e;r,\lambda) &= \sum_j \Pr\{\text{pattern } j\} \\
&\quad \times \frac{1}{2}\left[\Pr\{z < 0 \,|\, \Delta\phi_j, U_j, W_j\}\right. \\
&\qquad \left. + \Pr\{z > 0 \,|\, -\Delta\phi_j, U_j, W_j\}\right] \\
&= \sum_j \Pr\{\text{pattern } j\}\, P(e;r,\lambda \,|\, \Delta\phi_j, U_j, W_j). \qquad (4.1\text{-}15)
\end{aligned}
$$

In view of (4.1-13a) the probability that z is less than zero is known to be expressible in terms of Marcum's Q-function and the $I_0(\cdot)$ Bessel function [14]:

$$
\begin{aligned}
\Pr\{z < 0 \,|\, \Delta\phi, U, W\} &= \frac{1}{2}\left[1 - Q(\sqrt{b_1}, \sqrt{a_1}) + Q(\sqrt{a_1}, \sqrt{b_1})\right] \\
&\quad + \frac{1}{2}\,\frac{c_1 - c_2}{c_1 + c_2}\,\exp\left\{-\frac{a_1 + b_1}{2}\right\} I_0(\sqrt{a_1 b_1}), \qquad (4.1\text{-}16a)
\end{aligned}
$$

where

$$
a_1 \triangleq \frac{c_2 d_2(\Delta\phi)}{c_1 + c_2} = \frac{1}{1-r^2}\left\{U - rW\cos\Delta\phi - \sqrt{1-r^2}\,W \sin\Delta\phi\right\} \qquad (4.1\text{-}16b)
$$

and

$$b_1 \triangleq \frac{c_1 d_1(\Delta\phi)}{c_1+c_2} = \frac{1}{1-r^2}\{U - rW\cos\Delta\phi + \sqrt{1-r^2}W\sin\Delta\phi\}\ . \qquad (4.1\text{-}16c)$$

Similarly,

$$\Pr\{z > 0|-\Delta\phi,U,W\}$$

$$= \frac{1}{2}[1 - Q(\sqrt{b_0},\sqrt{a_0}) + Q(\sqrt{a_0},\sqrt{b_0})]$$

$$+ \frac{1}{2}\frac{c_2-c_1}{c_1+c_2}\exp\{-\frac{a_0+b_0}{2}\}I_0(\sqrt{a_0 b_0}), \qquad (4.1\text{-}17a)$$

with

$$a_0 \triangleq \frac{c_1 d_1(-\Delta\phi)}{c_1+c_2} = \frac{c_2 d_2(\Delta\phi)}{c_1+c_2} = a_1, \qquad (4.1\text{-}17b)$$

and

$$b_0 \triangleq \frac{c_2 d_2(-\Delta\phi)}{c_1+c_2} = \frac{c_1 d_1(\Delta\phi)}{c_1+c_2} = b_1. \qquad (4.1\text{-}17c)$$

Because $a_0 = a_1$ and $b_0 = b_1$, the terms containing the exponential in (4.1-16a) and (4.1-17a) cancel when they are averaged to get the conditional BER, resulting in

$$P(e\,;\,r,\lambda|\Delta\phi,U,W) = \frac{1}{2}[1 - Q(\sqrt{b_1},\sqrt{a_1}) + Q(\sqrt{a_1},\sqrt{b_1})]. \qquad (4.1\text{-}18)$$

#### 4.1.3.1 Comparison with Other Analyses

Note that the expression (4.1-18) does not depend on the cross-quadrature correlation coefficient $\lambda$, so this quantity can be assumed equal to zero for convenience. This same kind of symmetry (for equally probable data symbols) has been noted for DPSK [13], [16] with respect to r, the same-quadrature correlation coefficient.

The conditional binary FM error probability given by (4.1-18) can be shown [7, p 85] to be identical with the expressions given in [11]:

$$P(e;\, r,0|\Delta\phi,\, U,\, W) = \frac{1}{2}\,[1 - \sqrt{1-\beta^2/\alpha^2}\; Ie(\beta/\alpha,\alpha)], \tag{4.1-19a}$$

where $Ie(\cdot,\cdot)$ is the Rice function, with

$$\alpha = (a_1+b_1)/2\;, \qquad \beta = \sqrt{a_1 b_1}. \tag{4.1-19b}$$

Calculations of (4.1-19) involve numerical integration, since an equivalent expression is [11]

$$P(e;\, r,\, 0\,|\,\Delta\phi,\, U,\, W) = \frac{\sqrt{\alpha^2-\beta^2}}{2\pi}\int_0^{\pi} d\theta\, \frac{\exp[\,-(\alpha-\beta\cos\theta)]}{\alpha-\beta\,\cos\theta}\;. \tag{4.1-20}$$

Accurate calculation of the BER using (4.1-18) involving Marcum's Q-function is considered easier because of the possibility of a singularity in the integrand of (4.1-20).

It is noteworthy that the error expression (4.1-19a) represents the general form for the BER for any modulation scheme for which the receiver output at the sampling instant can be written as

$$z = R\,\cos(\phi_1 \pm \phi_2),\; R > 0, \tag{4.1-21}$$

with $\phi_1$ and $\phi_2$ distributed as the phase of a sinusoid in noise. This observation, due to Jain [17], is applied by him (with our parameters $r=\lambda=0$ and $\Delta\phi=\pi/2$) to undistorted detection of (a) hard-limited PSK with a perfect reference, (b) PSK with a noisy reference, (c) DPSK with differential detection ($\rho_1=\rho_2$), (d) binary FM with discriminator detection and without integrate-and-dump output filtering ($\sigma_1\neq\sigma_2$), and (e) BFSK.

4.1.3.2 A Useful Approximation

The noncentral chi-squared distribution with $\nu$ degrees of freedom and noncentrality parameter d is well approximated by ([18],[19])

$$\Pr\{\chi^2 > x|\nu,d\} \doteq Q(\sqrt{x-(\nu-1)/2} - \sqrt{d+(\nu-1)/2}), \qquad (4.1\text{-}22)$$

where $Q(\cdot)$ (with one argument) is the Gaussian complementary distribution function. Therefore, Marcum's Q-function is approximated by

$$Q(\alpha,\beta) = \Pr\{\chi^2 > \beta^2|2,\alpha^2\}$$

$$\doteq Q(\sqrt{\beta^2 - 1/2} - \sqrt{\alpha^2 + 1/2}) \qquad (4.1\text{-}23a)$$

$$\simeq Q(\beta - \alpha). \qquad (4.1\text{-}23b)$$

The conditional BER (4.1-18) then is approximated by

$$P(e;r,0|\Delta\phi,U,W) \simeq Q(\sqrt{b_1} - \sqrt{a_1}). \qquad (4.1\text{-}24)$$

For differential detection of narrowband digital FM, averaging the conditional BER over bit patterns gives the unconditional error probability

$$P(e) = \frac{1}{8}\{[P(e|000) + P(e|111)]$$
$$+ 2[P(e|011) + P(e|100)]$$
$$+ [P(e|010) + P(e|101)]\}. \qquad (4.1\text{-}25)$$

To evaluate the accuracy of the approximation given by (4.1-24), and to compare the exact formula (4.1-18) with the results presented in [11], we use the sample case of $h = 0.7$, $D = 1.0$, and $\lambda = 0$. The exact values for U, W, and $\Delta\phi$ needed to calculate $a_1$ and $b_1$ according to (4.1-16) were given in Table 2.1-3, and the same-quadrature correlation coefficient equals $e^{-\tau} = .0432$. Since in [11] the parameters for the 011 and 100 patterns are approximated, we shall use the approximate formulas

$$U\,(011) \simeq \frac{1}{2}[U(111) + U(010)], \qquad (4.1\text{-}26a)$$

$$W\,(011) \simeq \sqrt{U(111)\,U(010)}, \qquad (4.1\text{-}26b)$$

and

$$\Delta\phi\,(011) \simeq \frac{1}{2}[\Delta\phi\,(111) + \Delta\phi(010)]. \qquad (4.1\text{-}26c)$$

The BER for this example case is shown plotted against $E_b/N_0$ in Figure 4.1-2. These results indicate that the approximation tends to give a low estimate of the BER, but is quite good for high SNR.

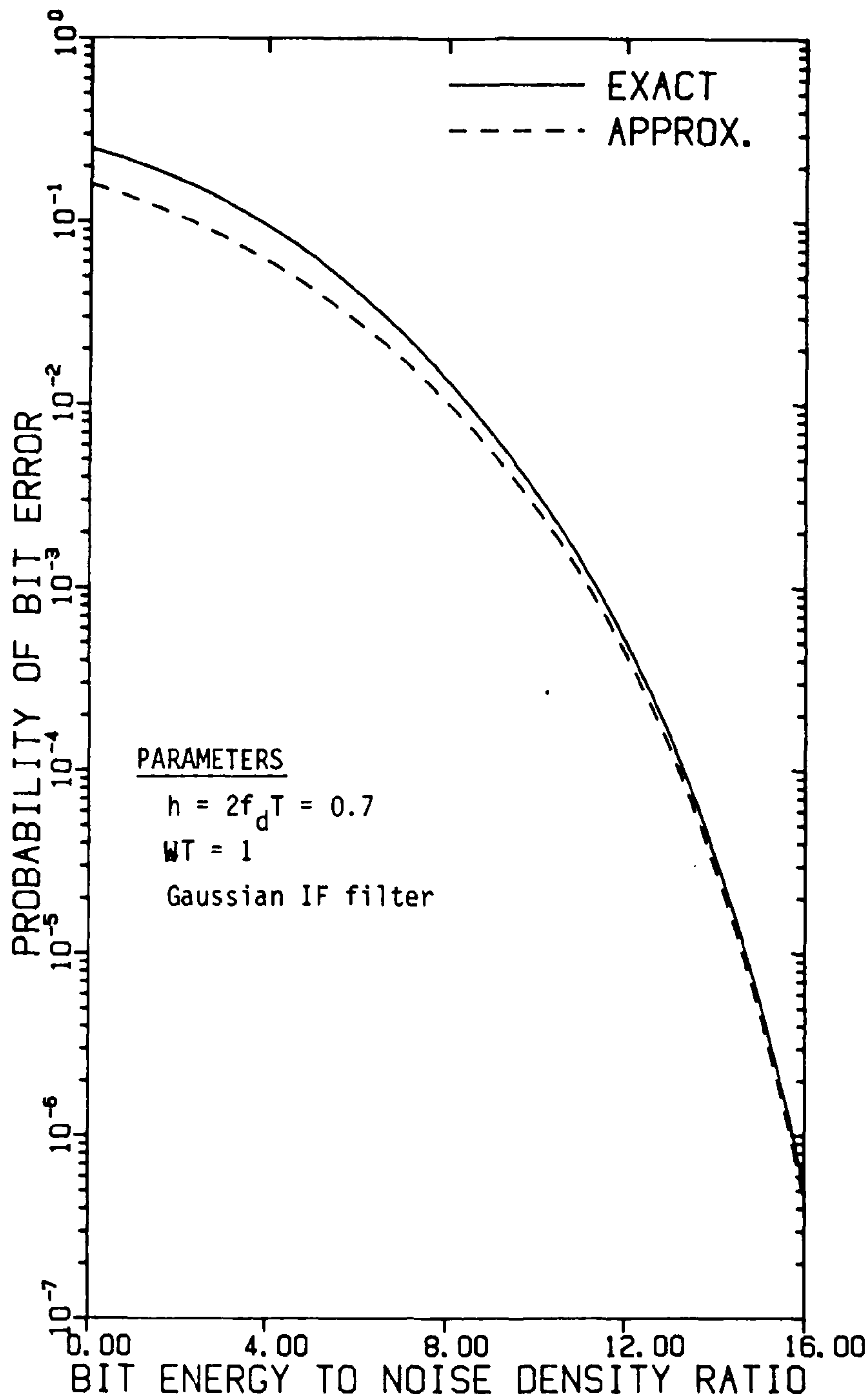



FIGURE 4.1-2 EXAMPLE CALCULATION OF NARROWBAND DIGITAL FM PERFORMANCE USING DIFFERENTIAL DETECTION

### 4.1.4 BER for L = 1 With Jamming

The case of partial-band noise jamming for L = 1 hop/bit can be treated as an extension of the unjammed case in the following way:

$$P(e; \gamma) = (1 - \gamma)P_a + \gamma P_b, \tag{4.1-27}$$

in which

$$P_a = \text{pattern-averaged BER for CNR} = E_b/N_0 \tag{4.1-28a}$$

$$P_b = \text{pattern-averaged BER for CNR} = E_b/N_T, \tag{4.1-28b}$$

where the averaging is of the conditional BER given by (4.1-18) over the pattern-dependent quantities U, W, and $\Delta\phi$ as listed in Table 2.1-3.

Numerical calculations of the L = 1 jammed BER are included in the next section, and are computed as a special case of the L > 1 BER expression developed there.

## 4.2 ANALYSIS FOR DIVERSITY SUM

The differential detector BER performance for FH/CPFSK using diversity may be expressed as

$$P_L(e) = \frac{1}{4}\{P_L(e|111) + 2P_L(e|011) + P_L(e|010)\} , \tag{4.2-1}$$

where L is the number of hops per bit, and

$$P_L(e|xly) = \sum_{\ell=0}^{L} \binom{L}{\ell} (1-\gamma)^{L-\ell} \gamma^{\ell} P_L(e|xly,\ell). \tag{4.2-2}$$

In this expression, the possible partial-band jamming events are indexed by $\ell$, the number of hops jammed out of L for a particular jamming event. As a function of the jamming bandwidth fraction $\gamma$, the probability of the event is

$$\Pr\{\ell \text{ hops jammed}\} = \binom{L}{\ell} (1-\gamma)^{L-\ell} \gamma^{\ell} . \tag{4.2-3}$$

### 4.2.1 Derivation of Conditional Error Probability

The decision statistic for L hops/bit diversity is the sum of samples of the differential detector output. As was shown in Section 4.1, each sample is equivalent to the difference of two equally-scaled noncentral chi-squared random variables, when the cross-quadrature correlation coefficient $\gamma$ is zero:

$$z_k \sim c_i[\chi^2(2;d_{i1}) - \chi^2(2;d_{i2})] \; , \tag{4.2-4a}$$

where the jamming condition is denoted by

$$i = \begin{cases} 0, \text{ hop unjammed} \\ 1, \text{ hop jammed.} \end{cases} \tag{4.2-4b}$$

From Section 4.1, we have

$$c_0 = \frac{1}{4}\sigma_N^2\sqrt{1-r^2} \quad , \quad c_1 = Kc_0; \tag{4.2-5a}$$

with the definition

$$K \triangleq \frac{\sigma_T^2}{\sigma_N^2} = \frac{(E_b/N_0)}{(E_b/N_T)} \; . \tag{4.2-5b}$$

Since the probability of error is not affected by uniform scaling of the samples, we may simplify matters somewhat by using

$$c_0 = 1, \quad c_1 = K. \tag{4.2-5c}$$

In (4.2-4a), the noncentrality parameters $d_{1i}$ and $d_{2i}$ for a particular data pattern are given by

$$d_{01,02} = \frac{U_N - r\, W_N \cos\Delta\phi \pm \sqrt{1-r^2}\, W_N \sin\Delta\phi}{1 - r^2} \qquad (4.2\text{-}6a)$$

and

$$d_{11,12} = (d_{01}, d_{02})/K, \qquad (4.2\text{-}6b)$$

with U and W being SNR parameters and $\Delta\phi$ being the (distorted) differential phase in the absence of noise. The SNR is related to the bit energy-to-noise density ratio by

$$\xi_N = \frac{1}{LD} \cdot \frac{E_b}{N_0} \quad . \qquad (4.2\text{-}7)$$

Since chi-squared variables combine to form chi-squared variables with higher degrees of freedom, the diversity sum for $\ell$ hops jammed is distributed as

$$z = \sum_{k=1}^{L} z_k \sim \chi^2[2(L-\ell);(L-\ell)d_{01}] - \chi^2[2(L-\ell);(L-\ell)d_{02}]$$
$$+ K\chi^2(2\ell;\ell d_{11}) - K\chi^2(2\ell;\ell d_{12}). \qquad (4.2\text{-}8)$$

### 4.2.1.1 Characteristic Function for Sum

The characteristic function for a chi-squared random variable with 2n degrees of freedom and noncentrality parameter d is

$$\phi_{\chi^2}(\nu;n,d) = E\{e^{j\nu\chi^2}\} = \frac{1}{(1-2j\nu)^n} \exp\left\{\frac{j\nu d}{1-2j\nu}\right\} \qquad (4.2\text{-}9)$$

Therefore, the characteristic function for z is

$$\varphi_z(\nu) = \varphi_{\chi^2}[\nu;L-\ell,(L-\ell)d_{01}]\varphi_{\chi^2}[-\nu;L-\ell,(L-\ell)d_{02}]$$

$$\times \varphi_{\chi^2}[K\nu;\ell,\ell d_{11}]\varphi_{\chi^2}[-K\nu;\ell,\ell d_{12}]$$

$$= [1+4\nu^2]^{-(L-\ell)}\,[1+4K^2\nu^2]^{-\ell}$$

$$\times \exp\left\{\frac{-4\nu^2(L-\ell)A + 2j\nu(L-\ell)B}{1+4\nu^2} + \frac{-4K\nu^2\ell A + 2j\nu\ell B}{1+4K^2\nu^2}\right\} \qquad (4.2\text{-}10)$$

in which

$$A = d_{01} + d_{02} = 2(U_N - rW_N\cos\Delta\phi)/(1-r^2) \qquad (4.2\text{-}11a)$$

$$B = d_{01} - d_{02} = 2W_N\sin\Delta\phi/\sqrt{1-r^2}. \qquad (4.2\text{-}11b)$$

4.2.1.2 BER From Characteristic Function

The cumulative probability distribution for a random variable may be written in terms of its characteristic function as follows [20]:

$$\Pr\{z < Z\} = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty \frac{d\nu}{\nu}\,\mathrm{Im}\{\varphi_z(\nu)e^{-j\nu Z}\}. \qquad (4.2\text{-}12)$$

Therefore the BER is given by

$$P(e) = \Pr\{z < 0\} = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty \frac{d\nu}{\nu}\,\mathrm{Im}\{\varphi_z(\nu)\}. \qquad (4.2\text{-}13)$$

Application to the conditional probability of error leads to

$$P_L(e|x1y,\ell) = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty \frac{d\nu}{\nu}\ [1+4\nu^2]^{-(L-\ell)}\ [1+4K^2\nu^2]^{-\ell}$$

$$\times \exp\left\{\frac{-4\nu^2(L-\ell)A}{1+4\nu^2} + \frac{-4\nu^2K\ell A}{1+4K^2\nu^2}\right\}$$

$$\times \sin\left\{\frac{2\nu(L-\ell)B}{1+4\nu^2} + \frac{2\nu\ell B}{1+4K^2\nu^2}\right\}. \qquad (4.2\text{-}14)$$

The integral in (4.2-14) may be converted to one with a finite interval of integration by using the following change of variable:

$$\nu = \frac{1}{2}\ \frac{\cos\theta}{1+\sin\theta}, \qquad (4.2\text{-}15a)$$

for which

$$d\nu = -\frac{1}{2}\ \frac{d\theta}{1+\sin\theta}. \qquad (4.2\text{-}15b)$$

The resulting expression is

$$P_L(e|x1y,\ell) = \frac{1}{2} - \frac{1}{\pi}\int_{-\pi/2}^{\pi/2} \frac{d\theta}{\cos\theta}\left(\frac{1+\sin\theta}{2}\right)^L [1+(K^2-1)(1-\sin\theta)/2]^{-\ell}$$

$$\times \exp\left\{-\frac{A}{2}(1-\sin\theta)\left[L-\ell+\frac{2\ell K}{2+(K^2-1)(1-\sin\theta)}\right]\right\}$$

$$\times \sin\left\{\frac{B}{2}\cos\theta\left[L-\ell+\frac{2\ell}{2+(K^2-1)(1-\sin\theta)}\right]\right\}. \qquad (4.2\text{-}16)$$

### 4.2.2 Numerical Results for Differential Detection

The BER for the differential detection of L hops/bit FH/CPFSK in partial-band noise jamming was computed as

$$P_L(e) = \frac{1}{4}\{ P_L(e|111) + 2P_L(e|011) + P_L(e|010)\}, \tag{4.2-17}$$

where

$$P_L(e|\underline{\beta}) = \sum_{\ell=0}^{L} \binom{L}{\ell} (1-\gamma)^{L-\ell} \gamma^{\ell} P_L(e|\underline{\beta},\ell) \tag{4.2-18}$$

and the conditional probabilities $P_L(e|\underline{\beta},\ell)$ are given by (4.2-16), using

$$A = 2(U_N - rW_N \cos\Delta\phi)/(1 - r^2) \tag{4.2-19a}$$

$$B = 2W_N \sin\Delta\phi/\sqrt{1 - r^2}\ . \tag{4.2-19b}$$

The subscript "N" denotes that U and W are calculated using

$$CNR = \rho_N = \frac{1}{L}\cdot\frac{1}{D} E_b/N_0. \tag{4.2-19c}$$

The value of the parameter K in (4.2-16) is

$$K = (E_b/N_0)/(E_b/N_T). \tag{4.2-20}$$

For the results shown below, we have used $E_b/N_0$ = 15 dB and $D = W_{IF}T = 1$.

Figures 4.2-1 through 4.2-3 show the L = 1 performance of the differential detector as a function of $E_b/N_J$, and parametric in $\gamma$, the fraction of the hop band which is jammed. The figures differ in that h values of 0.70, 0.65, and 0.60 are used, respectively. This assortment of values for the modulation index was used because, as we have shown, h = 0.7 (and D = 1) are considered best values for no jamming, while in [11] h = 0.6 (and D = 0.75) are said to be best values for the jamming case.

In comparison with Figure 3.1-5, we first note that all three of the differential detection results for L = 1 exhibit a higher jammed BER than that obtained using discriminator detection. This is not surprising, since differential detection is known to be less effective than discriminator detection without jamming. The difference in performance for worst-case

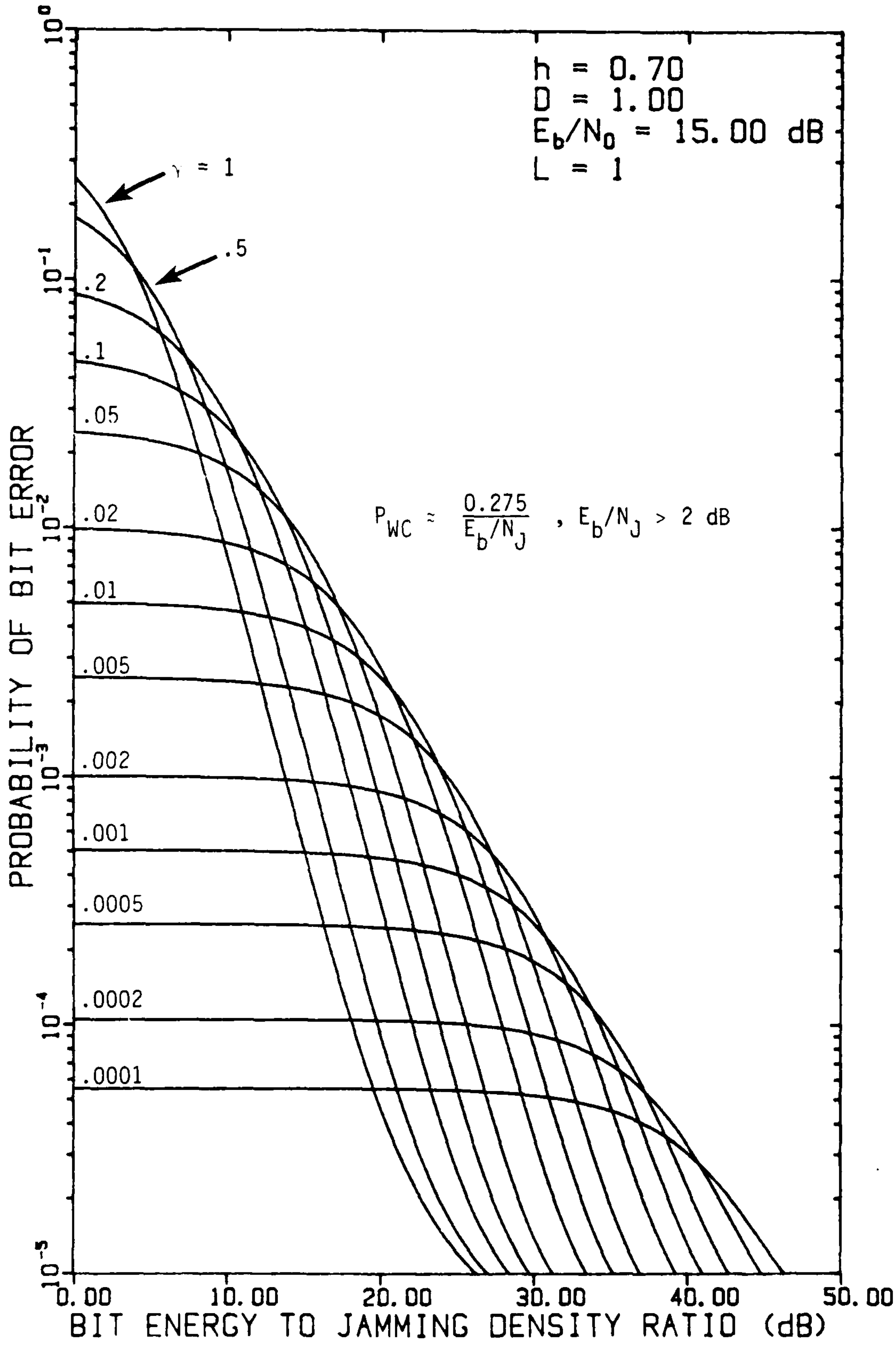



FIGURE 4.2-1 FH/CPFSK DIFFERENTIAL DETECTION PERFORMANCE IN PARTIAL-BAND NOISE JAMMING FOR L = 1 AND h = 0.70.

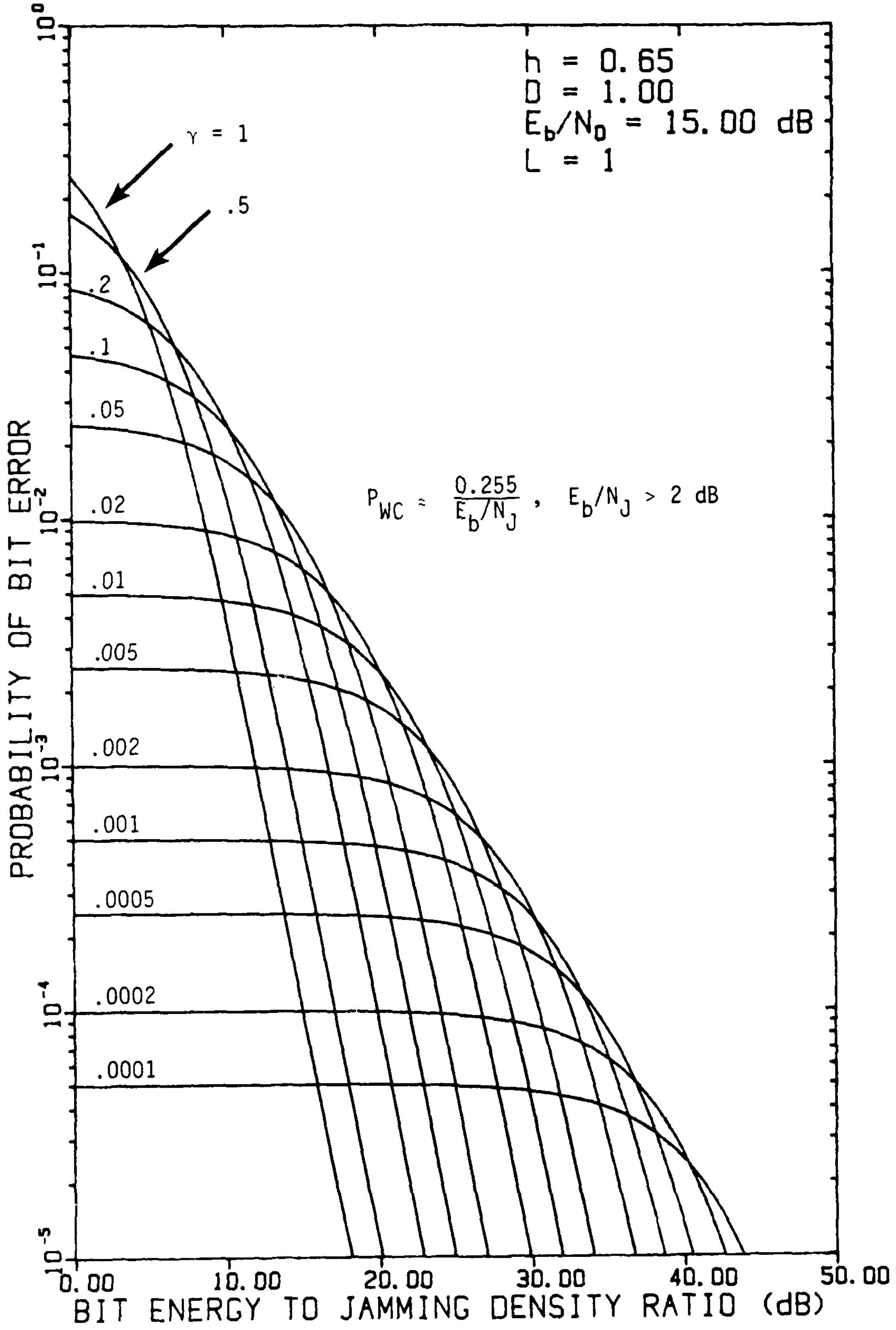



FIGURE 4.2-2 FH/CPFSK DIFFERENTIAL DETECTION PERFORMANCE IN PARTIAL-BAND NOISE JAMMING FOR L = 1 AND h = 0.65.

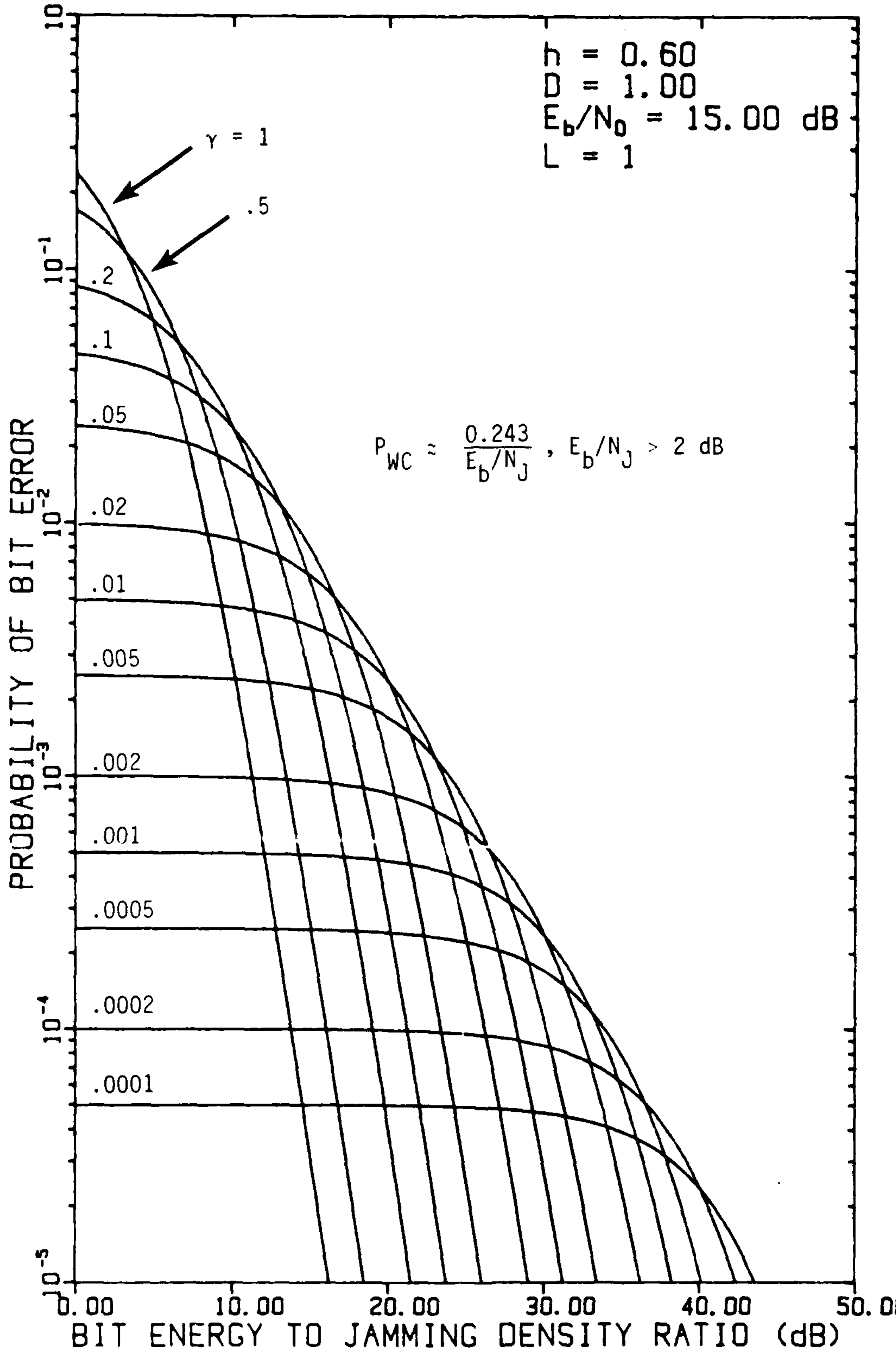



FIGURE 4.2-3 FH/CPFSK DIFFERENTIAL DETECTION PERFORMANCE IN PARTIAL-BAND NOISE JAMMING FOR L = 1 AND h = 0.60.

jamming is about 0.75 dB, for the parameter values used.

Observing now the differences among Figures 4.2-1, 4.2-2, and 4.2-3, we note that for h = 0.7, there is a definite "curling up" of the parametric error probability curves at the BER value of $10^{-5}$. This indicates that the noise-only error rate is between $10^{-6}$ and $10^{-5}$ for h = 0.7. As h is decreased to 0.65 or 0.6, the "curl" straightens out, indicating better performances in noise for these values. Thus for noise only, the BER for the differential detector is quite sensitive to the value of h. For example, for $\gamma = 1$, a $10^{-5}$ BER is obtained for the following total SNR values:

| | | | |
|---|---|---|---|
| h: | 0.7 | 0.65 | 0.6 |
| $E_b/N_T$: | 14.7 dB | 13.3 dB | 12.6 dB. |

This 2 dB spread in wide-band noise performance is not echoed in the worst-case jamming performance, however. The $P_{WC}$ relations shown in each figure reflect only a 0.54 dB spread in performance for the h values. Therefore, the value of h is not as critical in worst-case jamming as it is in Gaussian noise.

The jammed BER performances for diversity cases of L = 2 and L = 3 differential detector samples added are shown for h = 0.7 in Figures 4.2-4 and 4.2-5, respectively. The worst-case L = 2 performance is 0.36/0.275 = 1.2 dB worse than that for L =1; for L = 3, the worst-case performance is 0.50/0.275 = 2.6 dB worse. More significantly, these worst-case BER results are uniformly worse for variation in $E_b/N_J$, having the same inverse-linear dependence upon $E_b/N_J$ as the L = 1 case. Thus we have demonstrated that simple diversity summing of differentially-detected FH/CPFSK samples does not yield a diversity gain.

From [11] we know that diversity combining of hard decisions does

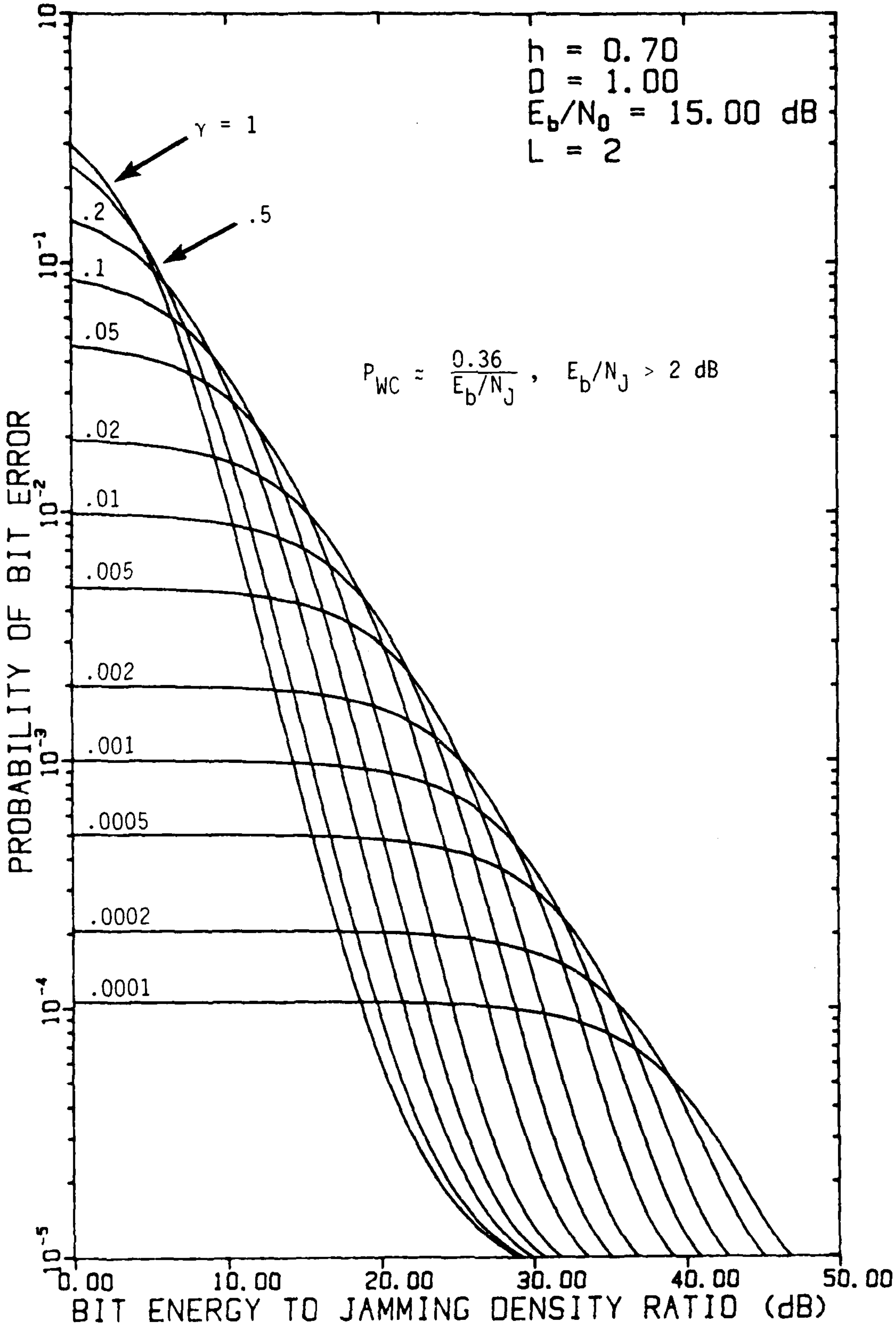



FIGURE 4.2-4 DIVERSITY SUM FH/CPFSK PERFORMANCE IN PARTIAL-BAND NOISE JAMMING FOR DIFFERENTIAL DETECTION AND L = 2.

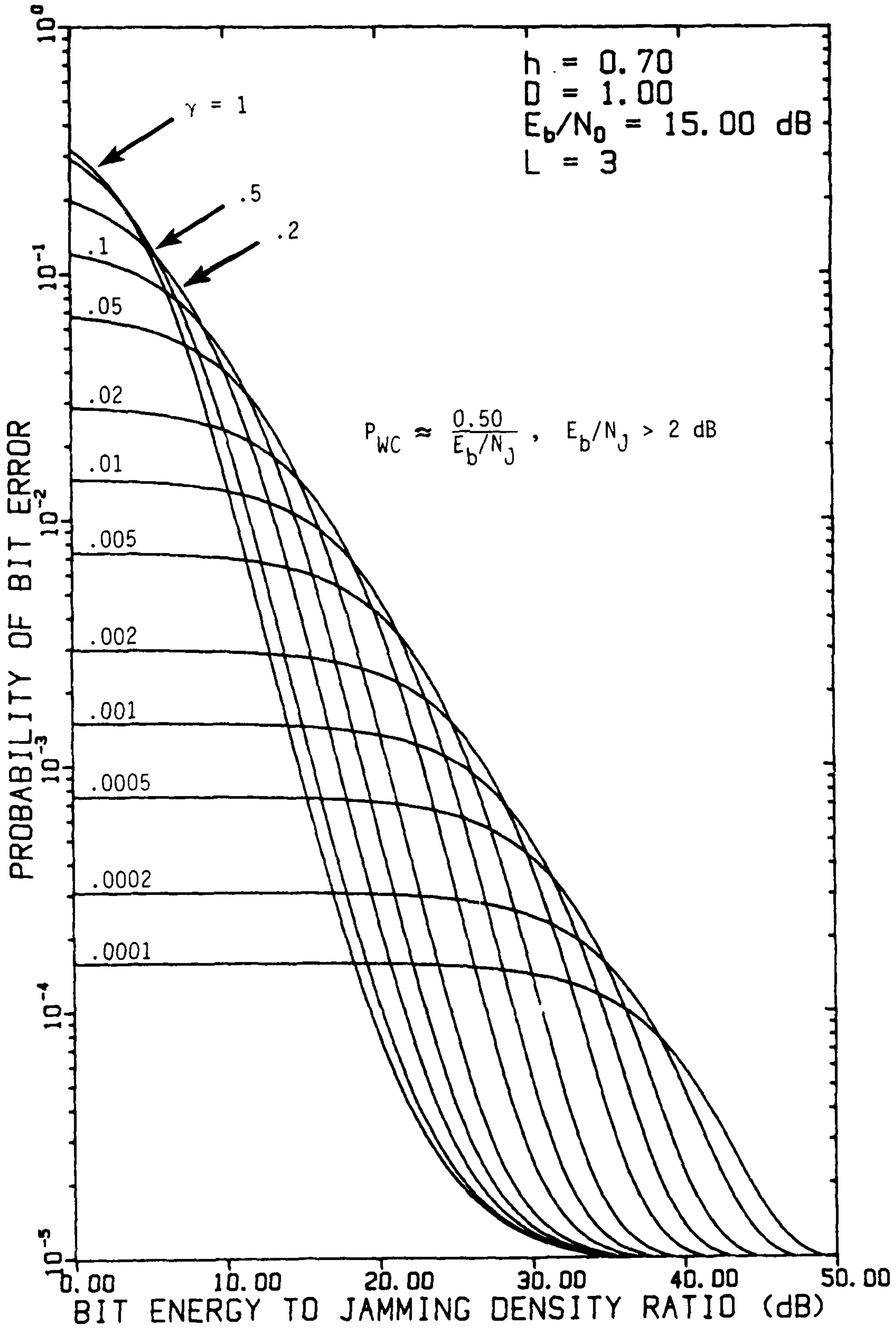



FIGURE 4.2-5 DIVERSITY SUM FH/CPFSK PERFORMANCE IN PARTIAL-BAND NOISE JAMMING FOR DIFFERENTIAL DETECTION AND L = 3.

produce a diversity gain for high $E_b/N_0$. Therefore, it is likely that forms of soft-decision combining exist which are better than hard-decision combining. We now know that summing of samples (soft decisions) is not one of them.

REFERENCES


[1] D.E. Cartier, "Limiter-Discriminator Detection Performance of Manchester and NRZ Coded FSK," IEEE Trans. on Aerospace and Electronic Syst., Vol. AES-13, pp. 62-70, January 1977.

[2] T.T. Tjhung and P.M. Wittke, "Carrier Transmission of Binary Data in a Restricted Band," IEEE Trans. Commun. Tech., Vol. COM-18, pp. 295-304, August 1970.

[3] R.F. Pawula, "On the Theory of Error Rates for Narrow-band Digital FM," IEEE Trans. Commun., Vol. COM-29, pp. 1634-1643, November 1981.

[4] D. Middleton, An Introduction to Statistical Communication Theory. New York: McGraw-Hill, 1960.

[5] I.S. Gradshteyn and I.M. Ryzhik, Table of Integrals, Series, and Products (4th ed.). New York: Academic, 1965.

[6] T. Mizuno, N. Morinaga, and T. Namekawa, "Transmission Characteristics of an M-ary Coherent PSK Signal Via a Cascade of N Bandpass Hard Limiters," IEEE Trans. on Commun., Vol. COM-24, No. 5, pp. 540-545, May 1976.

[7] J.H. Roberts, Angle Modulation: The Theory of Systems Assessment. Stevenage, Herts.: Peregrinus Ltd., 1977.

[8] R.F. Pawula, S.O. Rice, and J.H. Roberts, "Distribution of the Phase Angle Between Two Vectors Perturbed by Gaussian Noise," IEEE Trans. Commun., Vol. COM-30, pp. 1828-1841, August 1982.

[9] A. Weinberg, "Effects of a Hard Limiting Repeater on the Performance of a DPSK Data Transmission System," IEEE Trans. Commun., Vol. COM-25, No. 10, pp. 1128-1133, October 1977.

[10] J.S. Lee, R.H. French, and Y.K. Hong, "Error Performance of Differentially Coherent Detection of Binary DPSK Data Transmission on the Hard-Limiting Satellite Channel," IEEE Trans. Information Theory, Vol. IT-27, No. 4, pp. 489-497, July 1981.

[11] M.K. Simon and C.C. Wang, "Differential versus limiter-discriminator detection of narrow-band FM," IEEE Trans. Commun., Vol COM-31, pp 1227-1234, November 1983.

[12] L.E. Miller and J.S. Lee, "Bandpass correlator analysis for general input assumptions," IEEE Trans. Information Theory, Vol IT-28, pp 973-977, November 1982.

[13] S. Stein, "Unified analysis of certain coherent and noncoherent binary communications systems," IEEE Trans. Information Theory, Vol. IT-10, pp 43-51, January 1964.

[14] M. Schwartz, W.R. Bennett, and S. Stein, Communications Systems and Techniques. New York: McGraw-Hill, 1966.

[15] N. Ekanayake, "Binary DPSK transmission over terrestrial and satellite links," IEEE Trans. Information Theory, Vol IT-32, pp 125-129, January 1986.

[16] J.S. Lee, R.H. French, and Y.K. Hong, "Error performance of differentially coherent detection of binary DPSK data transmission on the hard-limiting satellite channel," IEEE Trans. Information Theory, Vol IT-27, pp 489-497, July 1981.

[17] P.C. Jain, "Error probabilities in binary angle modulation," IEEE Trans. Information Theory, Vol. IT-20, pp 36-42, January 1974.

[18] M. Sankaran, "Approximations to the non-central chi-square distribution," Biometrika, Vol 50 (1963), pp 199-204.

[19] L.E. Miller, "Computing R.O.C. for quadratic detectors," NSWC/WOL TR 76-148, October 1976 (AD-A050477).

[20] A.H. Nuttall, "Alternate forms for numerical evaluation of cumulative probability distributions directly from characteristic functions," Proceedings of the IEEE, Vol 58, pp 1872-1873, November 1970.

[21] D. J. Torrieri, "Frequency Hopping with Multiple Frequency-Shift Keying and Hard Decisions," IEEE Trans. Commun., Vol. COM-32, pp. 574-582, May 1984.

[22] M. K. Simon and C. C. Wang, "Limiter/Discriminator Detection of Narrowband FM in Jamming and Multipath Environments," Proc. IEEE Milit. Elec. Conf., paper 28.1, October 1982.

[23] L. E. Miller, J. S. Lee, and A. P. Kadrichu, "Probability of Error Analyses of a BFSK Frequency-Hopping System with Diversity under Partial-Band Jamming Interference--Part III: Performance of a Square-Law Self-Normalizing Soft-Decision Receiver," IEEE Trans. Commun., Vol. COM-34, pp. 669-675, July 1986.

[24] J. S. Lee, L. E. Miller, and R. H. French, "The Analyses of Uncoded Performances for Certain ECCM Receiver Design Strategies for Multi-hops/symbol FH/MFSK Waveforms," IEEE J. Select. Areas Commun., Vol. SAC-3, pp. 611-621, September 1985.

# APPENDIX A

## DERIVATION OF MODULO-2$\pi$ DIFFERENTIAL PHASE PDF BY DIRECT METHOD

From Section 2.1.5 of the text, the joint pdf of phases is

$$p_{\phi}(\phi_1,\phi_2) = \int_0^{\infty} dR_1 \int_0^{\infty} dR_2 \; p_{EP}(R_1,\phi_1,R_2,\phi_2) \tag{A-1}$$

where

$$p_{EP}(R_1,\phi_1,R_2,\phi_2) = \frac{R_1 R_2}{4\pi^2(1-\rho^2)} \exp\left\{\frac{-1}{1-\rho^2}\, Q_1\right\} \tag{A-2}$$

with

$$Q_1 = \frac{R_1^2+R_2^2}{2} + \alpha_1 + \alpha_2 - \rho R_1 R_2 \cos(\phi_1-\phi_2+\xi)$$

$$- XR_1 \cos(\phi_1-v) - YR_2 \cos(\phi_2-w)$$

$$- 2\rho\sqrt{\alpha_1\alpha_2} \cos(\phi_1-\phi_2+\xi). \tag{A-3}$$

The parameters $\rho$, $\xi$, X, v, Y, and w are defined in (2.1-37) of the text. Note that the convention we have adopted is that subscript "1" refers to quantities at time $t_1 = t_s$ and "2" refers to those at $t_2 = t_s - T$. Thus, for example, $\Delta\phi = \phi_1 - \phi_2$. This convention is opposite to that in [8].

## A.1 Transformation of Integrals

Our derivation begins with a transformation of variables. Let

$$\left.\begin{aligned} R_1 &= u\cos\left(\frac{\alpha}{2}+\frac{\pi}{4}\right)\sqrt{1-\mu^2} \\ R_2 &= u\cos\left(\frac{\alpha}{2}-\frac{\pi}{4}\right)\sqrt{1-\mu^2} \end{aligned}\right\} \qquad \begin{aligned} |\alpha| &< \pi/2 \\ u &> 0. \end{aligned} \tag{A-4}$$

The Jacobian is $u(1-\mu^2)/2$. With this transformation, we obtain

$$p_\phi(\phi_1,\phi_2) = \frac{(1-\mu^2)}{4}\cdot\frac{1}{4\pi^2}\int_{-\pi/2}^{\pi/2} d\alpha\, \cos\alpha \int_0^\infty du\, u^3 \exp\{-Q_2\}\ , \tag{A-5}$$

where $Q_2 = A\,u^2 - Bu + C$ (A-6a)

with $A = \frac{1}{2}\,[1 - \mu\cos\alpha\,\cos(\phi_1-\phi_2+\xi)]$ (A-6b)

$$B = \left(\sqrt{1-\mu^2}\right)^{-1}\left[X\cos\left(\frac{\alpha}{2}+\frac{\pi}{4}\right)\cos(\phi_1-v) + Y\cos\left(\frac{\alpha}{2}-\frac{\pi}{4}\right)\cos(\phi_2-w)\right]$$

$$= \frac{1}{\sqrt{2}}\ \frac{1}{\sqrt{1-\mu^2}}\sqrt{X^2+Y^2+2XY\cos\alpha\,\cos(\phi_1-\phi_2-v+w) + (Y^2-X^2)\sin\alpha}$$

$$\times\cos\left\{\phi_2 - \tan^{-1}\left[\frac{Y\cos(\frac{\alpha}{2}-\frac{\pi}{4})\sin w - X\cos(\frac{\alpha}{2}+\frac{\pi}{4})\sin(\phi_1-\phi_2-v)}{Y\cos(\frac{\alpha}{2}-\frac{\pi}{4})\cos w + X\cos(\frac{\alpha}{2}+\frac{\pi}{4})\cos(\phi_1-\phi_2-v)}\right]\right\} \tag{A-6c}$$

and

$$C = \frac{1}{1-\mu^2}\left[\rho_1 + \rho_2 - 2\mu\sqrt{\rho_1\rho_2}\,\cos(\phi_1-\phi_2+\xi)\right]. \tag{A-6d}$$

## A.2 Integration Over Unneeded Variable

We define the differential phase using the transformation of variables

$$x = \Phi_1 - \Phi_2 \qquad\qquad \Phi_1 = x + y$$
$$\text{or}$$
$$y = \Phi_2 \qquad\qquad \Phi_2 = y \tag{A-7}$$

with unit Jacobian. Integration of y over a 2π interval yields

$$p_{\Delta\phi}(x) = \frac{1-\rho^2}{8\pi} \int_{-\pi/2}^{\pi/2} d\alpha \; \cos\alpha \int_0^\infty du \; u^3 \; e^{-Au^2+C} \; I_0[u \cdot b(x)] \tag{A-8}$$

where

$$b(x) = \frac{1}{\sqrt{2}} \; \frac{1}{\sqrt{1-\rho^2}} \sqrt{X^2 + Y^2 + 2XY \cos\alpha \; \cos(x-v+w) + (Y^2-X^2) \sin\alpha} \; . \tag{A-9}$$

## A.3 Solution for Inner Integral

Using the integral

$$\int_0^\infty du \; u^3 \; e^{-Au^2} \; I_0(bu) = 2 \int_0^\infty dy \; y \; e^{-2Ay} \; I_0(b\sqrt{2y})$$

$$= \frac{2}{(2A)^2} \; {}_1F_1(2;1; \; \frac{b^2}{4A})$$

$$= \frac{2}{(2A)^2} \exp\left(\frac{b^2}{4A}\right) \left[1 + \frac{b^2}{4A}\right], \tag{A-10}$$

we obtain the pdf expression

$$p_{\Delta\phi}(x) = \frac{1-\mu^2}{4\pi} \int_{-\pi/2}^{\pi/2} d\alpha \, \frac{\cos\alpha}{(2A)^2} \, \exp\{-C + \frac{b^2}{4A}\}[1 + \frac{b^2}{4A}]. \qquad \text{(A-11)}$$

A.4 <u>Identification of Parameters</u>

From before,

$$2A = 1 - \mu \cos\alpha \cos(x+\xi)$$

$$= 1 - (r \cos x - \lambda \sin x)\cos\alpha. \qquad \text{(A-12)}$$

Now,

$$C - \frac{b^2}{4A} = \frac{1}{1-\mu^2} \Big| \rho_1 + \rho_2 - 2\mu\sqrt{\rho_1\rho_2} \cos(\Delta\phi + \xi)$$

$$- \frac{1}{4} \frac{X^2 + Y^2 + 2XY \cos\alpha \cos(x-v+w) + (Y^2-X^2)\sin\alpha}{1 - (r \cos x - \lambda \sin x) \cos\alpha} \Big|$$

$$= \frac{U + V \sin\alpha - W \cos\alpha \cos(x-\Delta\phi)}{1 - (r\cos x - \lambda \sin x)\cos\alpha} = E \qquad \text{(A-13)}$$

and

$$1 + \frac{b^2}{4A} = 1 - E + C$$

$$= 1 - E + 2 \frac{U - W(r \cos\Delta\phi - \lambda \sin\Delta\phi)}{1 - \mu^2}. \qquad \text{(A-14)}$$

Therefore

$$p_{\Delta\zeta}(x) = \frac{1-\mu^2}{4\pi} \int_{-\pi/2}^{\pi/2} d\alpha \, \frac{\cos\alpha \; e^{-E}}{[1-(r\cos x - \lambda\sin x)\cos\alpha]^2} \left[ 1-E+2\,\frac{U-W(r\cos\Delta\phi-\lambda\sin\Delta\phi)}{1-\mu^2} \right]. \quad \text{(A-15)}$$

Finally, changing the sign of the integration variable $\alpha$, and that of $\lambda$, gives the same expression as (2.1-55). The first sign change is arbitrary. The second reflects a different defining convention for $\lambda$.

# APPENDIX B

## A GENERAL PURPOSE PROGRAM FOR THE CPFSK ERROR PROBABILITY

The FORTRAN-77 program listed below computes the CPFSK bit error probability, including clicks, under the following assumptions:

(a) Given post-I.F. SNR (CNR), modulation index $h = 2f_d T$, and filter bandwidth-time product $D = W_{IF}T$; Gaussian-shaped spectrum.

(b) Only intersymbol interference effects due to adjacent bits are significant; eye pattern (differential phase) components contain only harmonics with frequencies $\leq R_b = 1/T$.

In addition to additive white Gaussian noise, the receiver is assumed to be subject to Gaussian noise interference with Gaussian-shaped spectrum with given SIR, bandwidth, and center frequency defined <u>prior</u> to the receiver filter. The bandwidth of this interference is specified by the input parameter

$$\beta \stackrel{\Delta}{=} W_I/W_{IF}, \tag{B-1}$$

and its frequency location relative to the signal carrier is specified by the parameter

$$\delta \stackrel{\Delta}{=} 2(f_I - f_c)T. \tag{B-2}$$

The program computes the error using the F function given by (2.1-56), and exploits symmetries to calculate

$$P(e) = \frac{1}{4}\,(P_0 + 2P_1 + P_2), \tag{B-3}$$

where

$$P_i = \frac{1}{2}\,[P(e|\overline{x}_i 0 \overline{y}) + P(e|x_i 1 y_i)]$$

$$= p_{ci}[F(0;\Delta\phi_i) - \frac{1}{2}\,F(\Delta\phi_i - \pi;\Delta\phi_i)$$

$$+ \frac{1}{2}\,F(\pi - \Delta\phi_i;\, -\Delta\phi_i)] + 1 - p_{ci}; \tag{B-4}$$

and $$p_{ci} = \exp\{-|\overline{N}_{ci}|\}\ . \tag{B-5}$$

The patterns are specified by

$$\begin{aligned} x_0 &= y_0 = 1 \\ x_1 &= 0,\ y_1 = 1 \\ x_2 &= y_2 = 0, \end{aligned} \tag{B-6}$$

and the pattern-dependent parameters are listed in Tables 2.1-1 and 2.1-3 of the text.

For no jamming or partial-band jamming, we use the input parameters SIR $\gg 1$, $\delta = 0$, and $\beta \gg 1$, with the SNR in the program being $E_b/N_0D$ when not jammed, or $E_b/N_TD$ when jammed.

```
PDP-11 FORTRAN-77 V4.0-1        16:34:08    24-Nov-87           Page 1
SPOTJAM.FTN;11          /F77/TR:BLOCKS/WR

0001            PROGRAM JAMSPO
        C
        C P(E) CALCULATIONS FOR SPOT JAMMING OF CPFSK
        C
        C J. S. LEE ASSOCIATES, INC.
        C 2001 JEFFERSON DAVIS HWY., SUITE 601
        C ARLINGTON, VA  22202
        C
        C ANALYSIS: L.E. MILLER    PROGRAM: R.H. FRENCH
        C 24 NOVEMBER 1987
        C
        C THIS PROGRAM CREATES TWO OUTPUTS:
        C       (A) A PRINT FILE WITH P(E) VS. SNR TABLES
        C       (B) BINARY FILES WITH PARAMETERS CODED INTO THE FILE
        C           NAMES FOR USE BY A PLOT GENERATION PROGRAM.  THESE
        C           ARE IN SINGLE PRECISION AND THE LOGS OF THE P(E)
        C           ARE TAKEN, FOR COMPATIBILITY WITH PLOT PACKAGE.
        C
        C           FILE NAME FORMAT:  xijjddbbb.fhh
        C
        C               WHERE:  x = 'W','X','Y','Z','A'....
        C                           FOR INTEGER PART OF D =
        C                            0,  1,  2,  3,  4, ...
        C                       i = 'A',...,'J' IF SIR .GE. 0  AND
        C                           TENTHS DIGIT OF D = 0,...,9
        C                           'M',...,'U' IF SIR .LT. 0  AND
        C                           TENTHS DIGIT OF D = 0,...,9
        C                      jj = INTEGER PART OF ABS(SIR), ROUNDED
        C                      dd = INTEGER PART OF 100*DELTA, ROUNDED
        C                     bbb = (1) IF BETA .LT. 1, 1000*BETA
        C                           (2) IF 1 .LE. BETA .LT. 999.5, BETA
        C                           (3) IF 999.5 .LE. BETA, 100*ALOG10(BETA)
        C                       f = INDICATOR OF CASE OF bbb: (1) S, (2) B, (3) L
        C                      hh = 100*H, ROUNDED
        C
        C               IN ALL CASES FIELDS ARE WRITTEN WITH LEADING ZEROS
        C
        C
        C PROGRESS MESSAGES ARE GIVEN TO THE TERMINAL IN FORM OF
        C WRITING OUT THE VARIOUS DO-LOOP INDICES
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETER (PI=3.14159265358979323846268D0)
0004            CHARACTER*13 FNAME
0005            CHARACTER*1 SSIR, PLUS, MINUS, EXT, BIGBET, SMABET, LOGBET,
             $           LET , LET2, FLTR
0006            REAL*4 PELOG(151), SNRDB(151)
0007            COMMON /PARMS/ H, HLST(4), NH, D, DLST(5), ND, DELLST(5), NDEL,
             $                BETLST(5), NBET,
             $                SNRLOW, SNRINC, NSNR, SIRLOW, SIRINC, NSIR
0008            COMMON /CONST/ A0, A1, A2, A3, A4,
             $                C1, C2, C3, C4, C5, C6, C7, C8, PIH
```

```
PDP-11 FORTRAN-77 V4.0-1        16:34:08    24-Nov-87           Page 2
SPOTJAM.FTN;11          /F77/TR:BLOCKS/WR

0009            COMMON /VARS/ RHO, R, ALAM
        C COMMON BLOCK USED IN CONJUNCTION WITH SUBROUTINE TO SIMULATE
        C AN INTEGER-->CHARACTER FUNCTION
0010            COMMON /TRICK/ LET1, LET2
0011            DATA PLUS, MINUS/'A','M'/
        C ENCODE RANGE OF BETA AS FIRST CHARACTER OF THE FILE EXTENSION
0012            DATA BIGBET, SMABET, LOGBET/'B','S','L'/
        C FILES ARE EXPLICITLY OPENED AND CLOSED AROUND EACH I/O SO AS
        C TO FORCE RESULTS TO BE WRITTEN TO DISK.  THIS PERMITS RESTART
        C WITH LOSS OF ONLY THE POINT IN PROGRESS WHEN THE SYSTEM IS SHUT DOWN.
0013            OPEN(UNIT=6,FILE='FOR006.DAT',STATUS='NEW',FORM='FORMATTED',
             $       ACCESS='SEQUENTIAL')
        C INSTALLATION-DEPENDENT OUTPUT HEADER PAGE
0014            CALL JSLGO
0015            CLOSE(UNIT=6)
        C SUPPRESS FLOATING UNDERFLOW MESSAGES
0016            CALL UFLOFF
        C INTERACTIVE INPUT OF PARAMETER SETS FOR THIS RUN
0017            CALL GET
        C
        C LOOP ON D
        C
0018            DO 1200 ID=1,ND
0019            D=DLST(ID)
0020            JD1=D
0021            JD2=(D-JD1)*10.D0
        C
        C LOOP ON H
        C
0022            DO 1000 IH=1,NH
0023            H=HLST(IH)
0024            JH=100.D0*H+0.5D0
        C
        C CALCULATE QUANTITIES NEEDED BY INTEGRANDS; PRECOMPUTED
        C AND PASSED THROUGH COMMON FOR EFFICIENCY
        C
0025            PIH=PI*H
0026            A0=EXP(-PIH*H/(8.D0*D*D))
0027            A1=EXP(-PI/(8.D0*D*D))
0028            A2=EXP(-PI/(2.D0*D*D))
0029            A3=EXP(-PI/(32.D0*D*D))
0030            A4=EXP(-9.D0*PI/(32.D0*D*D))
0031            C1=(4.D0*H/PI)*DCOS(PIH/2.D0)*A1/(1.D0-H*H)
0032            C2=DSINC(H/2.D0)
0033            C3=(4.D0*H/PI)*DSIN(PIH/2.D0)*A2/(4.D0-H*H)
0034            DENO=1.D0-4.D0*H*H
0035            IF(DENO.NE.0.D0) THEN
0036              C4=(8.D0*H/PI)*DCOS(PIH)*A3/DENO
0037            ELSE
        C L'HOSPITAL'S RULE WHEN THE DENOMINATOR VANISHES (H=1/2)
0038              C4=A3
0039            END IF
```

```
        C5=(8.DO*H/PI)*DCOS(PIH)*A4/(9.DO-4.DO*H*H)
        C6=DSINC(H)
        C7=C6*2.DO*H*H*A1/(1.DO-H*H)
        C8=C6*2.DO*H*H*A2/(4.DO-H*H)
C
C LOOP ON SIGNAL TO INTEREFERENCE RATIO
C
        DO 999 ISIR=1,NSIR
        SIRDB=SIRLOW+(ISIR-1)*SIRINC
        SIR=10.DO**(SIRDB/10.DO)
C ENCODING OF SIR FOR FILE NAME
        JSIR=DABS(SIRDB)+0.5DO
        IF(SIRDB.GE.0.DO) THEN
          SSIR=PLUS
        ELSE
          SSIR=MINUS
        END IF
        DO 900 IDEL=1,NDEL
        DELTA=DELLST(IDEL)
        JDEL=100.DO*DELTA+0.5DO
        DO 800 IBET=1,NBET
        BETA=BETLST(IBET)
C ENCODING OF BETA FOR FILE NAME
        IF(BETA.LT.1.DO) THEN
          JBET=1000.DO*BETA+0.5DO
          EXT=SMABET
        ELSE IF(BETA.LT.999.5DO) THEN
          JBET=BETA+0.5DO
          EXT=BIGBET
        ELSE
          JBET=DLOG10(BETA)*100.DO+0.5DO
          EXT=LOGBET
        END IF
C COMMON EXPRESSIONS INVOLVING BETA
        OBSQ=1.DO+BETA*BETA
        SOBSQ=DSQRT(OBSQ)
        EBD=DEXP(-PI*BETA*BETA*D*D/OBSQ)
C COSINE IS LOCALLY DEVELOPED ROUTINE TO INSURE COS(0)=1.000000000000
        COSDEL=COSINE(PI*DELTA/OBSQ)
        SINDEL=DSIN(PI*DELTA/OBSQ)
        ALPHA=DEXP(-PI*DELTA*DELTA*0.25DO/(D*D*OBSQ))/SOBSQ
C
C ENCODE THE UNITS DIGIT OF  D  AS AN INCREMENT TO THE FIRST
C LETTER OF THE FILE NAME
C
        FLTR='W'
        ITRICK=ICHAR(FLTR)+JD1
C IF IT GOES BEYOND 'Z', WRAP IT BACK TO 'A'
        IF(ITRICK.GT.90) ITRICK=ITRICK-25
C USE A COMMON BLOCK TO GET THE MODIFIED VALUE BACK AS A CHARACTER
        CALL FOOLIT(ITRICK)
        FLTR=LET1
```



```
C
C ENCODE THE TENTHS DIGIT OF  D  AS AN INCREMENT TO THE SECOND
C LETTER OF THE FILE NAME, WHICH ALSO ENCODES THE SIGN OF SIR
C
        ITRICK=ICHAR(SSIR)+JD2
C IF IT GOES BEYOND 'Z', WRAP IT BACK TO 'A'
        IF(ITRICK.GT.90) ITRICK=ITRICK-25
C USE A COMMON BLOCK TO GET THE MODIFIED VALUE BACK AS A CHARACTER
        CALL FOOLIT(ITRICK)
        SSIR=LET1
C CREATE NAME FOR BINARY RESULTS FILE
        WRITE(FNAME,10) FLTR,SSIR,JSIR,JDEL,JBET,EXT,JH
   10   FORMAT(2A1,I2.2,I2.2,I3.3,'.',A1,I2.2)
C CREATE THE BINARY RESULTS FILE AND WRITE TO IT A HEADER
C RECORD WITH RUN PARAMETERS
        OPEN(UNIT=1,FILE=FNAME,STATUS='NEW',FORM='UNFORMATTED',
     $       ACCESS='SEQUENTIAL')
        WRITE(1) H, D, SIRDB, DELTA, BETA, SNRLOW, SNRINC, NSNR
        CLOSE(UNIT=1)
C PRINTOUT HEADER
        OPEN(UNIT=6,FILE='FOR006.DAT',STATUS='OLD',FORM='FORMATTED',
     $       ACCESS='APPEND')
        WRITE(6,20) H, D, BETA, DELTA, SIRDB
        CLOSE(UNIT=6)
   20   FORMAT(' SPOT JAMMING OF CPFSK'/' H = ',F4.2,5X,'D = ',F4.2,5X,
     $         'BETA = ',F9.3,5X,'DELTA = ',F5.3/' SIR = ',
     $         F6.2,' dB'//' SNR (DB)',8X,'P(E)')
C
C LOOP ON SIGNAL TO NOISE RATIO
C
        DO 700 ISNR=1,NSNR
        SNRDB(ISNR)=SNRLOW+(ISNR-1)*SNRINC
        SNR=10.DO**(SNRDB(ISNR)/10.DO)
C PROGRESS MESSAGE TO TERMINAL
        WRITE(5,40) IH, ISIR, IBET, IDEL, ISNR
   40   FORMAT(' IH=',I1,'  ISIR=',I3,'  IBET=',I1,
     $         '  IDEL=',I1,'  ISNR=',I3)
C COMMON SUB-EXPRESSIONS INVOLVING SNR
        ARATIO=ALPHA*ALPHA*SNR/SIR
        RHO=SNR/(1.DO+ARATIO)
        R=(DEXP(-PI*D*D)+ARATIO*EBD*COSDEL)/(1.DO+ARATIO)
C ALAM IS MATHEMATICAL SYMBOL "LAMBDA"
        ALAM=ARATIO*EBD*SINDEL/(1.DO+ARATIO)
C COMPUTE RESULTS FOR THE DIFFERENT PHASE RELATIONSHIPS,
C MAKING USE OF SYMMETRIES TO AVOID REDUNDANT CALCULATIONS
        CALL PEE0(P0)
        CALL PEE1(P1)
        CALL PEE2(P2)
C TOTAL ERROR PROBABILITY
        PE=0.25DO*(P0+2.DO*P1+P2)
C
C
```

```
C TAKE LOG FOR SEMILOG PLOT; FLAG NON-POSITIVE AS -39 (WELL OFF
C THE PLOT LIMITS)
      IF(PE.GT.0.D0) THEN
        PELOG(ISNR)=DLOG10(PE)
      ELSE
        PELOG(ISNR)=-39.
      END IF
C WRITE THE LATEST RESULT TO THE BINARY DATA FILE
      OPEN(UNIT=1,FILE=FNAME,STATUS='OLD',FORM='UNFORMATTED',
     $     ACCESS='APPEND')
      WRITE(1) SNRDB(ISNR), PELOG(ISNR)
      CLOSE(UNIT=1)
C ... AND TO THE PRINT FILE
      OPEN(UNIT=6,FILE='FOR006.DAT',STATUS='OLD',FORM='FORMATTED',
     $     ACCESS='APPEND')
      WRITE(6,50) SNRDB(ISNR), PE
      CLOSE(UNIT=6)
   50 FORMAT(1X,F6.2,5X,1PD11.4)
  700 CONTINUE
  800 CONTINUE
  900 CONTINUE
  999 CONTINUE
 1000 CONTINUE
 1200 CONTINUE
      STOP 'DONE'
      END
```



```
      SUBROUTINE GET
C
C SUBROUTINE FOR INTERACTIVE INPUT OF PARAMETER SET FOR RUN
C
C       VALUES DISPLAYED IN SQUARE BRACKETS ARE DEFAULTS
C      WHICH ARE TAKEN IF JUST A CARRIAGE RETURN IS ENTERED.
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION DELDFL(5), BETDFL(5), DD(5)
      CHARACTER*10 FIELD, BLANKS
      COMMON /PARMS/ H, HLST(4), NH, D, DLST(5), ND, DELLST(5), NDEL,
     $                BETLST(5), NBET,
     $                SNRLOW, SNRINC, NSNR, SIRLOW, SIRINC, NSIR
      DATA DELDFL/0.D0, 0.35D0, 0.7D0, 0.8D0, 1.D0/
      DATA BETDFL/0.2D0, 0.1D0, 0.05D0, 0.01D0, 0.005D0/
      DATA DD/1.0D0, 1.1D0, 1.2D0, 0.8D0, 0.9D0/
      DATA BLANKS/'          '/
  100 WRITE(5,101)
  101 FORMAT(' HOW MANY VALUES OF H? [4]: ',$)
      READ(5,102,ERR=100) NH
  102 FORMAT(BN,I1)
      IF(NH.EQ.0) NH=4
      IF(NH.LT.0 .OR. NH.GT.4) GOTO 100
      DO 188 IN=1,NH
      DH=0.5D0+0.1D0*(IN-1)
    1 WRITE(5,2) IN, DH
    2 FORMAT('   ENTER H(',I1,') [',F3.1,']: ',$)
      READ(5,3,ERR=1) HLST(IN)
    3 FORMAT(BN,F4.0)
      IF(HLST(IN).EQ.0.D0) HLST(IN)=DH
  188 CONTINUE
  103 WRITE(5,104)
  104 FORMAT(' HOW MANY VALUES OF D? [1]: ',$)
      READ(5,105) ND
  105 FORMAT(I1)
      IF(ND.EQ.0) ND=1
      IF(ND.LT.0 .OR. ND.GT.5) GOTO 103
      DO 194 IN=1,ND
  106 WRITE(5,107) IN, DD(IN)
  107 FORMAT('   ENTER D(',I1,') [',F3.1,']: ',$)
      READ(5,108,ERR=106) DLST(IN)
  108 FORMAT(BN,F5.0)
      IF(DLST(IN).EQ.0.D0) DLST(IN)=DD(IN)
      IF(DLST(IN).LT.0.D0) GOTO 106
  194 CONTINUE
    7 WRITE(5,8)
    8 FORMAT(' HOW MANY DELTAS [3]: ',$)
      READ(5,9,ERR=7) NDEL
    9 FORMAT(BN,I1)
      IF(NDEL.EQ.0) NDEL=3
      IF(NDEL.LT.0 .OR. NDEL.GT.5) GOTO 7
C
```

```
          DO 14 I=1,NDEL
   10     WRITE(5,11) I, DELDFL(I)
   11     FORMAT('    ENTER DELTA(',I1,') [',F4.2,']: ',$)
          READ(5,12,ERR=10) FIELD
   12     FORMAT(A10)
          IF(FIELD.EQ.BLANKS) THEN
            DELLST(I)=DELDFL(I)
          ELSE
            READ(FIELD,13,ERR=10) DELLST(I)
   13       FORMAT(BN,F10.0)
          END IF
   14     CONTINUE
   15     WRITE(5,16)
   16     FORMAT(' HOW MANY BETAS? [4]: ',$)
          READ(5,17,ERR=15) NBET
   17     FORMAT(BN,I1)
          IF(NBET.EQ.0) NBET=4
          IF(NBET.LT.0 .OR. NBET.GT.5) GOTO 15
          DO 22 I=1,NBET
   18     WRITE(5,19) I,BETDFL(I)
   19     FORMAT('    ENTER BETA(',I1,') [',F5.3,']: ',$)
          READ(5,20,ERR=18) FIELD
   20     FORMAT(A10)
          IF(FIELD.EQ.BLANKS) THEN
            BETLST(I)=BETDFL(I)
          ELSE
            READ(FIELD,21,ERR=18) BETLST(I)
   21       FORMAT(BN,F10.0)
          END IF
   22     CONTINUE
   23     WRITE(5,24)
   24     FORMAT(' HOW MANY SNR''S? [151]: ',$)
          READ(5,241) NSNR
  241     FORMAT(BN,I2)
          IF(NSNR.EQ.0) NSNR=151
          IF(NSNR.LT.0 .OR. NSNR.GT.151) GOTO 23
  242     WRITE(5,243)
  243     FORMAT(' ENTER STARTING SNR IN dB [0.]: ',$)
          READ(5,244,ERR=242) FIELD
  244     FORMAT(A10)
          IF(FIELD.EQ.BLANKS) THEN
            SNRLOW=0.DO
          ELSE
            READ(FIELD,245,ERR=242) SNRLOW
  245       FORMAT(BN,F10.0)
          END IF
  246     WRITE(5,247)
  247     FORMAT(' ENTER INCREMENT OF SNR IN dB [0.1 dB]: ',$)
          READ(5,248,ERR=246) SNRINC
  248     FORMAT(BN,F10.0)
          IF(SNRINC.EQ.0.DO) SNRINC=0.1DO
      C
```



```
   27     WRITE(5,28)
   28     FORMAT(' HOW MANY SIRs? [1]: ',$)
          READ(5,29,ERR=27) NSIR
   29     FORMAT(BN,I3)
          IF(NSIR.EQ.0) NSIR=1
          IF(NSIR.LT.0) GOTO 27
   30     WRITE(5,31)
   31     FORMAT(' ENTER STARTING SIR IN dB [0.]: ',$)
          READ(5,32,ERR=30) SIRLOW
   32     FORMAT(BN,F7.0)
          IF(NSIR.GT.1) THEN
   33       WRITE(5,34)
   34       FORMAT(' ENTER INCREMENT FOR SIR IN dB [1.0]: ',$)
            READ(5,35,ERR=33) SIRINC
   35       FORMAT(BN,F7.0)
            IF(SIRINC.EQ.0.DO) SIRINC=1.DO
          ELSE
            SIRINC=0.DO
          END IF
          RETURN
          END
```

```
PDP-11 FORTRAN-77 V4.0-1        16:34:48    24-Nov-87           Page 9
SPOTJAM.FTN;11          /F77/TR:BLOCKS/WR

0001            SUBROUTINE PEEO(PO)
        C
        C COMPUTE PARTIAL RESULT PO
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETER (PI=3.14159265358979323846268D0)
0004            PARAMETER (HALFPI=1.57079632679489661923132D0)
0005            PARAMETER (HALMPI=-1.57079632679489661923132D0)
0006            EXTERNAL DGAU20, FARG
0007            DIMENSION WORK(15), STACK(15), HEAP(15)
0008            COMMON /PARMS/ H, HLST(4), NH, D, DLST(5), ND, DELLST(5), NDEL,
          $                     BETLST(5), NBET,
          $                     SNRLOW, SNRINC, NSNR, SIRLOW, SIRINC, NSIR
0009            COMMON /CONST/ AO, A1, A2, A3, A4,
          $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH
0010            COMMON /VARS/ RHO, R, ALAM
0011            COMMON /UVW/ UI, VI, WI, XLC, WLC
0012            UI=RHO*AO*AO
0013            VI=0.D0
0014            WI=UI
0015            DELPHI=PIH
        C FO(0,DELPHI)
0016            XLC=0.D0
0017            WLC=DELPHI
0018            CALL FSETUP
0019            CALL ADQUAD(HALMPI, HALFPI, FO, DGAU20, FARG, 1.D-6,
          $                 1.D-6, WORK, STACK, HEAP, 15, KODE)
0020            IF(KODE.NE.0) STOP 'PO: FO,1 FAILED TO CONVERGE'
0021            PSUM=FO+FO
        C FO(DELPHI-PI,DELPHI)
0022            XLC=DELPHI-PI
0023            WLC=DELPHI
0024            CALL FSETUP
0025            CALL ADQUAD(HALMPI, HALFPI, FO, DGAU20, FARG, 1.D-6,
          $                 1.D-6, WORK, STACK, HEAP, 15, KODE)
0026            IF(KODE.NE.0) STOP 'PO: FO,2 FAILED TO CONVERGE'
0027            PSUM=PSUM-FO
        C FO(-DELPHI+PI,-DELPHI)
0028            DELPHN=-DELPHI
0029            XLC=DELPHN+PI
0030            WLC=DELPHN
0031            CALL FSETUP
0032            CALL ADQUAD(HALMPI, HALFPI, FO, DGAU20, FARG, 1.D-6,
          $                 1.D-6, WORK, STACK, HEAP, 15, KODE)
0033            IF(KODE.NE.0) STOP 'PO: FO,3 FAILED TO CONVERGE'
0034            PSUM=PSUM+FO
0035            PSUM=0.125D0*PSUM/PI
        C CLICK TERM: NO BAR (CLOSED FORM RESULT AVAILABLE FOR THIS CASE)
0036            BARN=DEXP(-RHO*AO*AO)*H/2.D0
0037            PO=DEXP(-BARN)*(PSUM-1.D0)+1.D0
0038            RETURN
0039            END
```

```
PDP-11 FORTRAN-77 V4.0-1        16:34:52    24-Nov-87           Page 10
SPOTJAM.FTN;11          /F77/TR:BLOCKS/WR

0001            SUBROUTINE PEE1(P1)
        C
        C COMPUTE PARTIAL RESULT P1
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETER (PI=3.14159265358979323846268D0)
0004            PARAMETER (HALFPI=1.57079632679489661923132D0)
0005            PARAMETER (HALMPI=-1.57079632679489661923132D0)
0006            PARAMETER (TWOPI=6.28318530717958647692536D0)
0007            EXTERNAL DGAU20, FARG, BARN1
0008            DIMENSION WORK(15), STACK(15), HEAP(15)
0009            COMMON /PARMS/ H, HLST(4), NH, D, DLST(5), ND, DELLST(5), NDEL,
          $                     BETLST(5), NBET,
          $                     SNRLOW, SNRINC, NSNR, SIRLOW, SIRINC, NSIR
0010            COMMON /CONST/ AO, A1, A2, A3, A4,
          $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH
0011            COMMON /VARS/ RHO, R, ALAM
0012            COMMON /UVW/ UI, VI, WI, XLC, WLC
0013            UI=RHO*(0.5D0*DXI(C4+C5,2)+C7*C7+DXI(C6-C8,2))
0014            VI=RHO*(2.D0*C7*(C6-C8)-0.5D0*DXI(C4+C5,2))
0015            WI=DSQRT(UI*UI-VI*VI)
0016            DELPHI=DATAN2(C4+C5,C6-C7-C8)
        C FO(0,DELPHI)
0017            XLC=0.D0
0018            WLC=DELPHI
0019            CALL FSETUP
0020            CALL ADQUAD(HALMPI, HALFPI, FO, DGAU20, FARG, 1.D-6,
          $                 1.D-6, WORK, STACK, HEAP, 15, KODE)
0021            IF(KODE.NE.0) STOP 'P1: FO,1 FAILED TO CONVERGE'
0022            PSUM=FO+FO
        C FO(DELPHI-PI,DELPHI)
0023            XLC=DELPHI-PI
0024            WLC=DELPHI
0025            CALL FSETUP
0026            CALL ADQUAD(HALMPI, HALFPI, FO, DGAU20, FARG, 1.D-6,
          $                 1.D-6, WORK, STACK, HEAP, 15, KODE)
0027            IF(KODE.NE.0) STOP 'P1: FO,2 FAILED TO CONVERGE'
0028            PSUM=PSUM-FO
        C FO(-DELPHI+PI,-DELPHI)
0029            DELPHN=-DELPHI
0030            XLC=DELPHN+PI
0031            WLC=DELPHN
0032            CALL FSETUP
0033            CALL ADQUAD(HALMPI, HALFPI, FO, DGAU20, FARG, 1.D-6,
          $                 1.D-6, WORK, STACK, HEAP, 15, KODE)
0034            IF(KODE.NE.0) STOP 'P1: FO,3 FAILED TO CONVERGE'
0035            PSUM=PSUM+FO
0036            PSUM=0.125D0*PSUM/PI
        C CLICK TERM: NO BAR (BY NUMERICAL INTEGRATION)
0037            CALL ADQUAD(-1.D0, 0.D0, BARN, DGAU20, BARN1, 1.D-6,
          $                 1.D-6, WORK, STACK, HEAP, 15, KODE)
0038            IF(KODE.NE.0) STOP 'P1: BARN FAILED TO CONVERGE'
```

```
0039          P1=DEXP(-DABS(BARN)/TWOPI)*(PSUM-1.D0)+1.D0
0040          RETURN
0041          END
```



```
0001          SUBROUTINE PEE2(P2)
        C
        C COMPUTE PARTIAL RESULT P2
        C
0002          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003          PARAMETER (PI=3.14159265358979323846268D0)
0004          PARAMETER (HALFPI=1.57079632679489661923132D0)
0005          PARAMETER (HALMPI=-1.57079632679489661923132D0)
0006          PARAMETER (TWOPI=6.28318530717958647692536D0)
0007          EXTERNAL DGAU20, FARG, BARN2
0008          DIMENSION WORK(15), STACK(15), HEAP(15)
0009          COMMON /PARMS/ H, HLST(4), NH, D, DLST(5), ND, DELLST(5), NDEL,
         $                    BETLST(5), NBET,
         $                    SNRLOW, SNRINC, NSNR, SIRLOW, SIRINC, NSIR
0010          COMMON /CONST/ A0, A1, A2, A3, A4,
         $                    C1, C2, C3, C4, C5, C6, C7, C8, PIH
0011          COMMON /VARS/ RHO, R, ALAM
0012          COMMON /UVW/ UI, VI, WI, XLC, WLC
0013          UI=RHO*(C1*C1+DXI(C2-C3,2))
0014          VI=0.D0
0015          WI=UI
0016          DELPHI=2.D0*DATAN2(C1,C2-C3)
        C F0(0,DELPHI)
0017          XLC=0.D0
0018          WLC=DELPHI
0019          CALL FSETUP
0020          CALL ADQUAD(HALMPI, HALFPI, F0, DGAU20, FARG, 1.D-6,
         $                1.D-6, WORK, STACK, HEAP, 15, KODE)
0021          IF(KODE.NE.0) STOP 'P2: F0,1 FAILED TO CONVERGE'
0022          PSUM=F0+F0
        C F0(DELPHI-PI,DELPHI)
0023          XLC=DELPHI-PI
0024          WLC=DELPHI
0025          CALL FSETUP
0026          CALL ADQUAD(HALMPI, HALFPI, F0, DGAU20, FARG, 1.D-6,
         $                1.D-6, WORK, STACK, HEAP, 15, KODE)
0027          IF(KODE.NE.0) STOP 'P2: F0,2 FAILED TO CONVERGE'
0028          PSUM=PSUM-F0
        C F0(-DELPHI-PI,-DELPHI)
0029          DELPHN=-DELPHI
0030          XLC=DELPHN-PI
0031          WLC=DELPHN
0032          CALL FSETUP
0033          CALL ADQUAD(HALMPI, HALFPI, F0, DGAU20, FARG, 1.D-6,
         $                1.D-6, WORK, STACK, HEAP, 15, KODE)
0034          IF(KODE.NE.0) STOP 'P2: F0,3 FAILED TO CONVERGE'
0035          PSUM=PSUM+F0
0036          PSUM=0.125D0*PSUM/PI
        C CLICK TERM: NO BAR (BY NUMERICAL INTEGRATION)
0037          CALL ADQUAD(-1.D0, 0.D0, BARN, DGAU20, BARN2, 1.D-6,
         $                1.D-6, WORK, STACK, HEAP, 15, KODE)
0038          IF(KODE.NE.0) STOP 'P1: BARN FAILED TO CONVERGE'
```

```
PDP-11 FORTRAN-77 V4.0-1         16:34:58    24-Nov-87            Page 13
SPOTJAM.FTN;11            /F77/TR:BLOCKS/WR

0039            P2=DEXP(-DABS(BARN)/TWOPI)*(PSUM-1.D0)+1.D0
0040            RETURN
0041            END
```

```
PDP-11 FORTRAN-77 V4.0-1         16:35:04    24-Nov-87            Page 14
SPOTJAM.FTN;11            /F77/TR:BLOCKS/WR

0001            SUBROUTINE FSETUP
        C
        C SET UP CONSTANTS FOR F-INTEGRAND FUNCTION
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            COMMON /FCON/ WCOSWX, RCXLSX, WSINWX, RSXLCX
0004            COMMON /VARS/ RHO, R, ALAM
0005            COMMON /UVW/ UI, VI, WI, XLC, WLC
0006            WCOSWX=WI*COSINE(WLC-XLC)
0007            RCXLSX=R*COSINE(XLC)+ALAM*DSIN(XLC)
0008            WSINWX=WI*DSIN(WLC-XLC)
0009            RSXLCX=R*DSIN(XLC)-ALAM*COSINE(XLC)
0010            RETURN
0011            END
```

```
PDP-11 FORTRAN-77 V4.0-1         16:35:06    24-Nov-87            Page 15
SPOTJAM.FTN;11            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION FARG(Z)
        C
        C INTEGRAND FUNCTION FOR COMPUTING F0,1 BY NUMERICAL INTEGRATION
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            COMMON /FCON/ WCOSWX, RCXLSX, WSINWX, RSXLCX
0004            COMMON /UVW/ UI, VI, WI, XLC, WLC
0005            SINZ=DSIN(Z)
0006            COSZ=COSINE(Z)
0007            PART1=UI-VI*SINZ-WCOSWX*COSZ
0008            DENOM=1.D0-RCXLSX*COSZ
0009            EARG=PART1/DENOM
0010            FARG=DEXP(-EARG)*(WSINWX/PART1+RSXLCX/DENOM)
0011            RETURN
0012            END
```

```
PDP-11 FORTRAN-77 V4.0-1         16:35:08    24-Nov-87            Page 16
SPOTJAM.FTN;11            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION BARN1(X)
        C
        C INTEGRAND FUNCTION FOR CLICK TERM OF P1
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETER (PI=3.14159265358979323846268D0)
0004            PARAMETER (HALFPI=1.57079632679489661923132D0)
0005            PARAMETER (PI1R5=4.71238898038468985769396D0)
0006            PARAMETER (TWOPI=6.28318530717958647692536D0)
0007            COMMON /CONST/ A0, A1, A2, A3, A4,
          $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH
0008            COMMON /VARS/ RHO, R, ALAM
0009            X1R5=PI1R5*X
0010            XR5=HALFPI*X
0011            PIX=PI*X
0012            TPX=TWOPI*X
0013            UIX=C5*DSIN(X1R5)-C4*DSIN(XR5)
0014            VIX=C6+C7*COSINE(PIX)-C8*COS(TPX)
0015            WIX=PI1R5*COSINE(X1R5)-HALFPI*C4*COSINE(XR5)
0016            ZIX=TWOPI*C8*DSIN(TPX)-PI*C7*DSIN(PIX)
0017            D=UIX*UIX+VIX*VIX
0018            BARN1=DEXP(-RHO*D)*(WIX*VIX-ZIX*UIX)/D
0019            RETURN
0020            END
```

```
PDP-11 FORTRAN-77 V4.0-1         16:35:12    24-Nov-87            Page 17
SPOTJAM.FTN;11            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION BARN2(X)
        C
        C INTEGRAND FUNCTION FOR CLICK TERM OF P2
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETER (PI=3.14159265358979323846268D0)
0004            PARAMETER (TWOPI=6.28318530717958647692536D0)
0005            COMMON /CONST/ A0, A1, A2, A3, A4,
          $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH
0006            COMMON /VARS/ RHO, R, ALAM
0007            PIX=PI*X
0008            TPX=TWOPI*X
0009            UIX=C1*COSINE(PIX)
0010            VIX=C2-C3*COSINE(TPX)
0011            WIX=-PI*C1*DSIN(PIX)
0012            ZIX=TWOPI*C3*DSIN(TPX)
0013            D=UIX*UIX+VIX*VIX
0014            BARN2=DEXP(-RHO*D)*(WIX*VIX-ZIX*UIX)/D
0015            RETURN
0016            END
```

```
      SUBROUTINE ADQUAD(XL,XU,Y,QR,F,TOL,ABSTOL,
     1                  WORK,STACK,HEAP,N,KODE)
C
C ADAPTIVE QUADRATURE ALGORITHM FOR NUMERICAL INTEGRATION
C
C     XL - LOWER LIMIT OF INTEGRAL (IN)
C     XU - UPPER LIMIT OF INTEGRAL (IN)
C     Y - VALUE OF INTEGRAL (OUT)
C     QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C             WITH CALLING SEQUENCE
C             CALL QR(XL,XU,F,Y)
C      F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C    TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C ABSTOL - ABSOLUTE ERROR TOLERANCE
C   WORK - WORK ARRAY OF SIZE N (IN)
C  STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
C   HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
C          SAME ARRAY AS WORK (IN)
C      N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C   KODE - ERROR  NDICATOR (OUT)
C          0 -- NO ERROR
C          1 -- WORK ARRAYS TOO SMALL
C
C R. H. FRENCH, 14 AUGUST 1984
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      EXTERNAL F
      DIMENSION WORK(N),STACK(N),HEAP(N)
      KODE=0
      Y=0.D0
      WORK(1)=XU
      CALL QR(XL,XU,F,T)
      HEAP(1)=T
      A=XL
      NPTS=1
      EPS=TOL
      STACK(1)=EPS
   10 B=WORK(NPTS)
      XM=(A.B)*0.5D0
      CALL QR(A,XM,F,P1)
      CALL QR(XM,B,F,P2)
      TEST=DMAX1(EPS*DABS(T),ABSTOL)
      IF(DABS(T-P1-P2).LE.TEST .OR. DABS(T).LE.ABSTOL) GOTO 20
C SPLIT IT
      NPTS=NPTS+1
      IF(NPTS.GT.N) THEN
        Y=P1+P2
        KODE=1
        RETURN
      END IF
      WORK(NPTS)=XM
      HEAP(NPTS)=P2
```





```
      T=P1
      EPS=DMAX1(EPS/2.D0,5.D-16)
      STACK(NPTS)=EPS
      GOTO 10
C FINISHED A PIECE
   20 Y=Y+P1+P2
      EPS=STACK(NPTS)
      T=HEAP(NPTS)
      NPTS=NPTS-1
      A=B
      IF(NPTS.EQ.0) RETURN
      GOTO 10
      END
```



```
      SUBROUTINE FOOLIT(ITRICK)
C
C CONVERT AN INTEGER ARGUMENT INTO TWO CHARACTERS BY PUTTING
C IT INTO A COMMON BLOCK WHICH CONTAINS TWO CHARACTER*1 VARIABLES
C IN THE OTHER PROGRAM WHICH REFERENCES IT
C
      COMMON /TRICK/ JTRICK
      JTRICK=ITRICK
      RETURN
      END
```

```
PDP-11 FORTRAN-77 V4.0-1        09:55:15    25-Nov-87            Page 1
DGAU20.FTN;1              /F77/TR:BLOCKS/WR

          SUBROUTINE DGAU20(A,B,F,ANSWER)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C 20-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL C
C                                                     C
C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4 C
C                                                     C
C R. H. FRENCH, 21 JUNE 1983                          C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
          DIMENSION X(10),W(10)
          DATA X/ 0.076526521133497333755D0,
     1            0.227785851141645078080D0,
     1            0.373706088715419560673D0,
     1            0.510867001950827098004D0,
     1            0.636053680726515025453D0,
     1            0.746331906460150792614D0,
     1            0.839116971822218823395D0,
     1            0.912234428251325905868D0,
     1            0.963971927277913791268D0,
     1            0.993128599185094924786D0 /
          DATA W/ 0.152753387130725850698D0,
     1            0.149172986472603746788D0,
     1            0.142096109318382051329D0,
     1            0.131688638449176626898D0,
     1            0.118194531961518417312D0,
     1            0.101930119817240435037D0,
     1            0.083276741576704748725D0,
     1            0.062672048334109063570D0,
     1            0.040601429800386941331D0,
     1            0.017614007139152118312D0 /
          ANSWER=0.D0
          BMAO2=(B-A)/2.D0
          BPAO2=(B+A)/2.D0
          DO 10 I=1,10
          C=X(I)*BMAO2
          Y1=BPAO2+C
          Y2=BPAO2-C
          ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
   10     CONTINUE
          ANSWER=ANSWER*BMAO2
          RETURN
          END
```

# APPENDIX C

## A PROGRAM FOR THE CPFSK ERROR PROBABILITY WHEN L = 2 HOPS/BIT

```
0001          PROGRAM H2CFSK
        C
        C COMPUTE ERROR PROBABILITY, AVERAGED OVER DATA PATTERNS,
        C FOR L=2 HOPS CPFSK/FH (WITH INTERLEAVING).
        C
        C J.S. LEE ASSOCIATES, INC.
        C 2001 JEFFERSON DAVIS HWY., SUITE 601
        C ARLINGTON, VA  22202
        C
        C ANALYSIS: L.E. MILLER        PROGRAM: R.H. FRENCH
        C
        C 1 DEC 1987
        C
0002          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003          PARAMETER (PI=3.14159265358979323846268D0)
0004          CHARACTER*13 FNAME
0005          CHARACTER*1 EJECT
0006          EXTERNAL BCI111, BCI011, BCI010
0007          LOGICAL GLT1
0008          COMMON /PARMS/ EBNODB, GAMLST(20), NGAM, EBNJLO, EBNJIN, NEBNJ,
     $                       D, H
0009          COMMON /CONST/ A0, A1, A2, A3, A4,
     $                       C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $                       C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
0010          COMMON /THINGS/ ARE, DELPHI, OMRSQ2
0011          DATA EJECT/' '/
        C
        C INSTALLATION-DEPENDENT HEADER PAGE AND TURN OFF UNDERFLOW MESSAGES
        C
0012          OPEN(UNIT=6,FILE='FOR006.DAT',STATUS='NEW',FORM='FORMATTED',
     $            ACCESS='SEQUENTIAL')
0013          CALL JSLGO
0014          CLOSE(UNIT=6)
0015          CALL UFLOFF
0016          CALL GET
        C
        C COMPUTE CONSTANTS
        C
0017          EBNO=10.D0**(EBNODB/10.D0)
0018          RHON=0.5D0*EBNO/D
0019          PIH=PI*H
0020          JH=10.D0*H+0.5D0
0021          JD=10.D0*D+0.5D0
0022          JNO=EBNODB+0.5D0
0023          ARE=DEXP(-PI*D*D)
0024          OMRSQ2=(1.D0-ARE*ARE)/2.D0
0025          A0=EXP(-PIH*H/(8.D0*D*D))
0026          A1=EXP(-PI/(8.D0*D*D))
0027          A2=EXP(-PI/(2.D0*D*D))
0028          A3=EXP(-PI/(32.D0*D*D))
0029          A4=EXP(-9.D0*PI/(32.D0*D*D))
0030          C1=(4.D0*H/PI)*DCOS(PIH/2.D0)*A1/(1.D0-H*H)
```



```
0031          C1PM=-C1*PI
0032          C2=DSINC(H/2.D0)
0033          C3=(4.D0*H/PI)*DSIN(PIH/2.D0)*A2/(4.D0-H*H)
0034          C3P2=2.D0*PI*C3
0035          DENO=1.D0-4.D0*H*H
0036          IF(DENO.NE.0.D0) THEN
0037            C4=(8.D0*H/PI)*DCOS(PIH)*A3/DENO
0038          ELSE
        C L'HOSPITAL'S RULE WHEN THE DENOMINATOR VANISHES (H=1/2)
0039            C4=A3
0040          END IF
0041          C4PH=C4*PI/2.D0
0042          C5=(8.D0*H/PI)*DCOS(PIH)*A4/(9.D0-4.D0*H*H)
0043          C5P3R2=1.5D0*C5*PI
0044          C6=DSINC(H)
0045          C7=C6*2.D0*H*H*A1/(1.D0-H*H)
0046          C7P=C7*PI
0047          C8=C6*2.D0*H*H*A2/(4.D0-H*H)
0048          C8P2=C8*PI*2.D0
        C
        C COMPUTE QUANTITIES NOT DEPENDING ON GAMMA AND EB/NJ
        C   -- CLICK PROBABILITIES (UNJAMMED) .OR THE THREE DATA PATTERNS
        C
0049          CALL BARN(RHON, BCI111, BNC111)
0050          P11100=DEXP(-BNC111)
0051          P11101=BNC111*P11100
0052          CALL BARN(RHON, BCI011, BNC011)
0053          P01100=DEXP(-BNC011)
0054          P01101=BNC011*P01100
0055          CALL BARN(RHON, BCI010, BNC010)
0056          P01000=DEXP(-BNC010)
0057          P01001=BNC010*P01000
        C
        C  -- P1(0,0) AND P2(0,0) FOR THE THREE DATA PATTERNS
        C
0058          CALL CP111(RHON, RHON, CP1700, TRASH, CP2700)
0059          CALL CP011(RHON, RHON, CP1300, TRASH, CP2300)
0060          CALL CP010(RHON, RHON, CP1200, TRASH, CP2200)
        C
        C LOOP ON EB/NJ
        C
0061          DO 900 IEBNJ=1,NEBNJ
0062          EBNJDB=EBNJLO+(IEBNJ-1)*EBNJIN
0063          EBNJ=10.D0**(EBNJDB/10.D0)
0064          JJ=EBNJDB+0.5D0
        C
        C CONSTRUCT NAME FOR BINARY OUTPUT FILE, OPEN, AND WRITE HEADER
        C
0065          WRITE(FNAME,10) JH, JD, JNO, JJ
0066     10   FORMAT('TH',I1.1,I2.2,I2.2,I2.2,'.DAT')
0067          OPEN(UNIT=2, FILE=FNAME, STATUS='NEW',
     $            ACCESS='SEQUENTIAL', FORM='UNFORMATTED')
```

```
              WRITE(2) H, D, EBNODB, EBNJDB
              CLOSE(UNIT=2)
C PRINTED PAGE HEADER
              OPEN(UNIT=6, FILE='FOR006.DAT', STATUS='OLD',
     $             ACCESS='APPEND', FORM='FORMATTED')
              WRITE(6,20) EJECT, H, D, EBNODB, EBNJDB
   20         FORMAT(A1,'TWO-HOP CPFSK/FH'/1X,'H = ',F4.2,5X,'D = ',F4.2/
     $             1X,'EB/NO = ',F8.4,' dB'/1X,'EB/NJ = ',F5.2,' dB'//
     $             4X,'GAMMA',12X,'PE',11X,'PE(111)',9X,'PE(011)',
     $             9X,'PE(010)')
              CLOSE(UNIT=6)
              EJECT='1'
C
C LOOP ON GAMMA
C
              DO 800 IGAM=1,NGAM
              GAMMA=GAMLST(IGAM)
              OMG=1.DO-GAMMA
C
C TEST FOR GAMMA < 1 TO AVOID UNNECESSARY COMPUTATIONS FOR THE
C CASE WHEN GAMMA = 1 AND 1-GAMMA = 0
C
              GLT1=GAMMA.LT.1.DO
              GEBNJ=GAMMA*EBNJ
              EBNT=GEBNJ*EBNO/(GEBNJ+EBNO)
              RHOT=0.5DO*EBNT/D
C
C CLICK TERMS FOR JAMMED EVENTS
C
              CALL BARN(RHOT, BCI111, RESULT)
              P11110=DEXP(-RESULT)
              P11111=RESULT*P11110
              CALL BARN(RHOT, BCI011, RESULT)
              P01110=DEXP(-RESULT)
              P01111=RESULT*P01110
              CALL BARN(RHOT, BCI010, RESULT)
              P01010=DEXP(-RESULT)
              P01011=RESULT*P01010
C
C CONTINUOUS TERMS FOR PATTERN 111
C
              IF(GLT1) THEN
                CALL CP111(RHOT, RHON, CP1710, CP1701, CP2710)
              END IF
              CALL CP111(RHOT, RHOT, CP1711, TRASH, CP2711)
C
C CONTINUOUS TERMS FOR PATTERN 011
C
              IF(GLT1) THEN
                CALL CP011(RHOT, RHON, CP1310, CP1301, CP2310)
              END IF
              CALL CP011(RHOT, RHOT, CP1311, TRASH, CP2311)
```





```
C
C CONTINUOUS TERMS FOR PATTERN 010
C
              IF(GLT1) THEN
                CALL CP010(RHOT, RHON, CP1210, CP1201, CP2210)
              END IF
              CALL CP010(RHOT, RHOT, CP1211, TRASH, CP2211)
C
C COMPUTE P(E;I,J;PATTERN) FOR THE THREE PATTERNS
C --- PATTERN 111
C
              IF(GLT1) THEN
                PE700=1.DO - P11100*P11100 + P11100*P11100*CP2700
     $                     - P11101*P11100*CP1700 - P11100*P11101*CP1700
                PE710=1.DO - P11110*P11100 + P11110*P11100*CP2710
     $                     - P11111*P11100*CP1710 - P11110*P11101*CP1701
              END IF
              PE711=1.DO - P11110*P11110 + P11110*P11110*CP2711
     $                   - P11111*P11110*CP1711 - P11110*P11111*CP1711
C --- PATTERN 011
              IF(GLT1) THEN
                PE300=1.DO - P01100*P01100 + P01100*P01100*CP2300
     $                     - P01101*P01100*CP1300 - P01100*P01101*CP1300
                PE310=1.DO - P01110*P01100 + P01110*P01100*CP2310
     $                     - P01111*P01100*CP1310 - P01110*P01101*CP1301
              END IF
              PE311=1.DO - P01110*P01110 + P01110*P01110*CP2311
     $                   - P01111*P01110*CP1311 - P01110*P01111*CP1311
C --- PATTERN 010
              IF(GLT1) THEN
                PE200=1.DO - P01000*P01000 + P01000*P01000*CP2200
     $                     - P01001*P01000*CP1200 - P01000*P01001*CP1200
                PE210=1.DO - P01010*P01000 + P01010*P01000*CP2210
     $                     - P01011*P01000*CP1210 - P01010*P01001*CP1201
              END IF
              PE211=1.DO - P01010*P01010 + P01010*P01010*CP2211
     $                   - P01011*P01010*CP1211 - P01010*P01011*CP1211
C
C CONDITIONAL ERROR PROBABILITIES, GIVEN PATTERN
C
              P7=GAMMA*GAMMA*PE711
              P3=GAMMA*GAMMA*PE311
              P2=GAMMA*GAMMA*PE211
              IF(GLT1) THEN
                P7=OMG*OMG*PE700 + 2.DO*GAMMA*OMG*PE710 + P7
                P3=OMG*OMG*PE300 + 2.DO*GAMMA*OMG*PE310 + P3
                P2=OMG*OMG*PE200 + 2.DO*GAMMA*OMG*PE210 + P2
              END IF
C
C TOTAL ERROR PROBABILITY
C
              PE=0.25DO*(P7+2.DO*P3+P2)
```

PDP-11 FORTRAN-77 V4.0-1 15:50:31 3-Dec-87 Page 5
CPFSK2.FTN;32 /F77/TR:BLOCKS/WR

```
C
C WRITE THE ANSWERS
C
      OPEN(UNIT=2, FILE=FNAME, STATUS='OLD', ACCESS='APPEND',
     $     FORM='UNFORMATTED')
      WRITE(2) GAMMA, PE, P7, P3, P2
      CLOSE(UNIT=2)
      OPEN(UNIT=6, FILE='FOR006.DAT', STATUS='OLD', ACCESS='APPEND',
     $     FORM='FORMATTED')
      WRITE(6,30) GAMMA, PE, P7, P3, P2
   30 FORMAT(1X,1PD11.4,4(5X,1PD11.4))
      CLOSE(UNIT=6)
  800 CONTINUE
  900 CONTINUE
      STOP 'COMPUTATIONS COMPLETED'
      END
```

PDP-11 FORTRAN-77 V4.0-1 15:50:55 3-Dec-87 Page 6
CPFSK2.FTN;32 /F77/TR:BLOCKS/WR

```
      SUBROUTINE GET
C
C INTERACTIVE INPUT OF PARAMETERS FOR THE RUN
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*10 FIELD, BLANKS
      DIMENSION DGAM(20)
      COMMON /PARMS/ EBNODB, GAMLST(20), NGAM, EBNJLO, EBNJIN, NEBNJ,
     $              D, H
      DATA BLANKS/'          '/
      DATA DGAM/ 1.D-4, 2.D-4, 5.D-4, 1.D-3, 2.D-3, 5.D-3,
     $           1.D-2, 2.D-2, 5.D-2, 1.D-1, 2.D-1, 5.D-1,
     $           1.D00, 7.D-4, 7.D-3, 7.D-2, 7.D-1, 3.D-3,
     $           3.D-2, 3.D-1/
   10 WRITE(5,11)
   11 FORMAT(' ENTER EB/NO IN dB [11.7495 dB]: ',$)
      READ(5,12,ERR=10) FIELD
   12 FORMAT(A10)
      IF(FIELD.EQ.BLANKS) THEN
        EBNODB=11.7495D0
      ELSE
        READ(FIELD,13,ERR=10) EBNODB
   13   FORMAT(BN,F10.0)
      END IF
   20 WRITE(5,21)
   21 FORMAT(' ENTER H [0.7]: ',$)
      READ(5,22,ERR=20) H
   22 FORMAT(BN,F5.0)
      IF(H.EQ.0.D0) H=0.7D0
      IF(H.LT.0.D0 .OR. H.GE.1.D0) GOTO 20
   30 WRITE(5,31)
   31 FORMAT(' ENTER D [1.0]: ',$)
      READ(5,32,ERR=30) D
   32 FORMAT(BN,F5.0)
      IF(D.EQ.0.D0) D=1.D0
      IF(D.LT.0.D0) GOTO 30
   40 WRITE(5,41)
   41 FORMAT(' HOW MANY VALUES OF GAMMA? [13]: ',$)
      READ(5,42,ERR=40) NGAM
   42 FORMAT(BN,I2)
      IF(NGAM.EQ.0) NGAM=13
      IF(NGAM.LT.0 .OR. NGAM.GT.20) GOTO 40
      DO 59 IN=1,NGAM
   50 WRITE(5,51) IN, DGAM(IN)
   51 FORMAT('    ENTER GAMMA(',I2,') [',1PD7.1,']: ',$)
      READ(5,52,ERR=50) GAMLST(IN)
   52 FORMAT(BN,D12.0)
      IF(GAMLST(IN).EQ.0.D0) GAMLST(IN)=DGAM(IN)
      IF(GAMLST(IN).LT.0.D0 .OR. GAMLST(IN).GT.1.D0) GOTO 50
   59 CONTINUE
   60 WRITE(5,61)
   61 FORMAT(' ENTER STARTING VALUE OF EB/NJ IN dB [0.]: ',$)
```

```
PDP-11 FORTRAN-77 V4.0-1        15:50:55     3-Dec-87            Page 7
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

              READ(5,62,ERR=60) EBNJLO
        62    FORMAT(BN,F5.0)
        70    WRITE(5,71)
        71    FORMAT(' ENTER INCREMENT OF EB/NJ IN dB [10.]: ',$)
              READ(5,72,ERR=70) EBNJIN
        72    FORMAT(BN,F5.0)
              IF(EBNJIN.EQ.0.D0) EBNJIN=10.
        80    WRITE(5,81)
        81    FORMAT(' ENTER NUMBER OF VALUES OF EB/NJ [6]: ',$)
              READ(5,82,ERR=80) NEBNJ
        82    FORMAT(BN,I3)
              IF(NEBNJ.EQ.0) NEBNJ=6
              IF(NEBNJ.LT.0 .OR. NEBNJ.GT.151) GOTO 80
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        15:51:04     3-Dec-87            Page 8
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

              SUBROUTINE BARN(ROW, FNBAR, RESULT)
C
C COMPUTE ABSOLUTE VALUE OF AVERAGE CLICK NUMBER, BY
C NUMERICAL INTEGRATION OF THE SPECIFIED FUNCTION FNBAR
C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              PARAMETER (TWOPI=6.283185307179586476925360D0)
              DIMENSION WORK(15), STACK(15), HEAP(15)
              EXTERNAL FNBAR, DGAU20
              COMMON /ROSE/ RHO
              RHO=ROW
              CALL ADQUAD(-1.D0, 0.D0, BARNC, DGAU20, FNBAR, 1.D-5,
     $                    1.D-5, WORK, STACK, HEAP, 15, KODE)
              IF(KODE.NE.0) STOP 'CLICK TERM FAILED TO CONVERGE'
              RESULT=DABS(BARNC/TWOPI)
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        15:51:06     3-Dec-87            Page 9
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

              DOUBLE PRECISION FUNCTION BCI111(X)
C
C INTEGRAND FUNCTION FOR CLICK NUMBER FOR PATTERN 111
C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              COMMON /CONST/ A0, A1, A2, A3, A4,
     $                       C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $                       C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
              COMMON /ROSE/ RHO
              PHX=PIH*X
              UPRIME=A0*DSIN(PHX)
              VPRIME=A0*COSINE(PHX)
              W=PIH*VPRIME
              Z=-PIH*UPRIME
              DDD=UPRIME*UPRIME + VPRIME*VPRIME
              BCI111 = DEXP(-RHO*DDD) * (W*VPRIME-UPRIME*Z)/DDD
              RETURN
              END
```

```
      DOUBLE PRECISION FUNCTION BCI011(X)
C
C INTEGRAND FUNCTION FOR CLICK NUMBER FOR PATTERN 011
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI=3.14159265358979323846268D0)
      PARAMETER (TWOPI=6.28318530717958647692536D0)
      PARAMETER (HALFPI=1.57079632679489661923132D0)
      PARAMETER (PI1R5=4.71238898038468985769396D0)
      COMMON /CONST/ A0, A1, A2, A3, A4,
     $               C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $               C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
      COMMON /ROSE/ RHO
      HPX=HALFPI*X
      P15X=PI1R5*X
      TPX=TWOPI*X
      PIX=PI*X
      UPRIME=C4*DSIN(HPX) - C5*DSIN(P15X)
      VPRIME=C6 + C7*COSINE(PIX) - C8*COSINE(TPX)
      W=C4PH*COSINE(HPX) - C5P3R2*COSINE(P15X)
      Z=C8P2*DSIN(TPX) - C7P*DSIN(PIX)
      DDD=UPRIME*UPRIME + VPRIME*VPRIME
      BCI011 = DEXP(-RHO*DDD) * (W*VPRIME-UPRIME*Z)/DDD
      RETURN
      END
```



```
      DOUBLE PRECISION FUNCTION BCI010(X)
C
C INTEGRAND FUNCTION FOR CLICK NUMBER FOR PATTERN 010
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI=3.14159265358979323846268D0)
      PARAMETER (TWOPI=6.28318530717958647692536D0)
      COMMON /CONST/ A0, A1, A2, A3, A4,
     $               C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $               C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
      COMMON /ROSE/ RHO
      TPX=TWOPI*X
      PIX=PI*X
      UPRIME=C1*COSINE(PIX)
      VPRIME=C2-C3*COSINE(TPX)
      W=C1PM*DSIN(PIX)
      Z=C3P2*DSIN(TPX)
      DDD=UPRIME*UPRIME + VPRIME*VPRIME
      BCI010 = DEXP(-RHO*DDD) * (W*VPRIME-UPRIME*Z)/DDD
      RETURN
      END
```





```
      SUBROUTINE CP111(RHOI, RHOJ, P1IJ, P1JI, P2)
C
C COMPUTE CONDITIONAL PROBABILITIES:
C                                        P1(I,J;111)
C                                        P1(J,I;111)
C                                    AND P2(I,J;111)
C
C     NOTE: I=J IS TESTED TO AVOID UNNECESSARY RECOMPUTATION
C           OF P1JI
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI=3.14159265358979323846268D0)
      PARAMETER (HALFPI=1.57079632679489661923132D0)
      PARAMETER (TWOPI=6.28318530717958647692536D0)
      DIMENSION WORK(15), STACK(15), HEAP(15)
      EXTERNAL DGAU20, QF
      LOGICAL EQUAL
      COMMON /CONST/ A0, A1, A2, A3, A4,
     $               C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $               C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
      COMMON /THINGS/ ARE, DELPHI, OMRSQ2
      COMMON /QPAR/ UQ, VQ, WQ, TERM
      COMMON /FPAR/ UF, VF, WF
C
C TEST FOR EQUALITY OF RHOI AND RHOJ
C
      EQUAL=RHOI.EQ.RHOJ
C
C SET DELTA PHI
C
      DELPHI=PIH
C
C COMMON SUB-EXPRESSIONS FOR THE FI AND FJ
C
      UN=A0*A0
      VN=0.D0
      WN=UN
C
C FI(DELPHI+PI)
C
      UF=RHOI*UN
      VF=RHOI*VN
      WF=RHOI*WN
      FIPLUS=FI(DELPHI+PI)
C
C FI(PI-DELPHI)
C
      FIMINU=FI(PI-DELPHI)
C
C FI(DELPHI-PI)
C
      FIUNIM=FI(DELPHI-PI)
```

```
C
C IF NEEDED, FI(-DELPHI)
C
      IF(DELPHI.GT.HALFPI) THEN
        FIMDP=FI(-DELPHI)
      END IF
C
C IF NEEDED FOR P1(J,I), COMPUTE INTEGRALS OF QJ*FI
C
      IF(.NOT.EQUAL) THEN
        UQ=RHOJ*UN
        VQ=RHOJ*VN
        WQ=RHOJ*WN
        TERM=1.D0+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
        CALL ADQUAD(PI-DELPHI, TWOPI-DELPHI, AJI, DGAU20, QF, 1.D-5,
     $               1.D-5, WORK, STACK, HEAP, 15, KODE)
        IF(KODE.NE.0) STOP 'P1(J,I!111) PARTA DID NOT CONVERGE'
        CALL ADQUAD(TWOPI-DELPHI,DELPHI+PI, BJI, DGAU20, QF, 1.D-5,
     $               1.D-5, WORK, STACK, HEAP, 15, KODE)
        IF(KODE.NE.0) STOP 'P1(J,I!111) PARTB DID NOT CONVERGE'
      END IF
C
C FJ(DELPHI+PI)
C
      UF=RHOJ*UN
      VF=RHOJ*VN
      WF=RHOJ*WN
      FJPLUS=FI(DELPHI+PI)
C
C FJ(DELPHI-PI)
C
      FJMINU=FI(DELPHI-PI)
C
C IF RHOI .NE. RHOJ, NEED FJ(PI-DELPHI) AND POSSIBLY FJ(-DELPHI)
C
      IF(.NOT.EQUAL) THEN
        FJUNIM=FI(PI-DELPHI)
        IF(DELPHI.GT.HALFPI) FJMDP=FI(-DELPHI)
      END IF
C
C THE INTEGRAL TERMS
C     (1) SET UP THE PARAMETERS FOR QI(X) FUNCTION
C
      UQ=UN*RHOI
      VQ=VN*RHOI
      WQ=WN*RHOI
      TERM=1.D0+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
C
C     (2) DO THE INTEGRATION FOR P1IJ
C
      CALL ADQUAD(PI-DELPHI, TWOPI-DELPHI, PARTA, DGAU20, QF, 1.D-5,
     $             1.D-5, WORK, STACK, HEAP, 15, KODE)
```





```
      IF(KODE.NE.0) STOP 'P1(I,J!111) PARTA DID NOT CONVERGE'
      CALL ADQUAD(TWOPI-DELPHI,DELPHI+PI, PARTB, DGAU20, QF, 1.D-5,
     $             1.D-5, WORK, STACK, HEAP, 15, KODE)
      IF(KODE.NE.0) STOP 'P1(I,J!111) PARTB DID NOT CONVERGE'
C
C    (3) RETURN THE FUNCTION P1IJ
C
      P1IJ=FJPLUS*(FIPLUS-FIMINU)-PARTA-PARTB
      IF(DELPHI.GT.HALFPI) P1IJ=P1IJ+FJPLUS+FIPLUS-FIMDP
C
C    (4) DO THE INTEGRATION FOR P2
C
      CALL ADQUAD(DELPHI-PI, PI-DELPHI, RESULT, DGAU20, QF, 1.D-5,
     $             1.D-5, WORK, STACK, HEAP, 15, KODE)
      IF(KODE.NE.0) STOP 'P2(I,J!111) DID NOT CONVERGE'
C
C    (5) RETURN P2
C
      P2=RESULT-FJMINU*(FIMINU-FIUNIM)
      IF(HALFPI.GT.DELPHI) P2=P2+FIDMP-FIUNIM-FJMINU
C
C    (6) RETURN P1JI
C
      IF(EQUAL) THEN
        P1JI=P1IJ
      ELSE
        P1JI=FIPLUS*(FJPLUS-FJUNIM)-AJI-BJI
        IF(DELPHI.GT.HALFPI) P1JI=P1JI+FIPLUS+FJPLUS-FJMDP
      END IF
      RETURN
      END
```

```
      SUBROUTINE CP011(RHOI, RHOJ, P1IJ, P1JI, P2)
C
C COMPUTE CONDITIONAL PROBABILITIES:
C                                         P1(I,J;011)
C                                         P1(J,I;011)
C                                     AND P2(I,J;011)
C     NOTE: I=J IS TESTED TO AVOID UNNECESSARY RECOMPUTATION
C           OF P1JI
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI=3.14159265358979323846268D0)
      PARAMETER (HALFPI=1.57079632679489661923132D0)
      PARAMETER (TWOPI=6.28318530717958647692536D0)
      DIMENSION WORK(15), STACK(15), HEAP(15)
      EXTERNAL DGAU20, QF
      LOGICAL EQUAL
      COMMON /CONST/ A0, A1, A2, A3, A4,
     $               C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $               C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
      COMMON /THINGS/ ARE, DELPHI, OMRSQ2
      COMMON /QPAR/ UQ, VQ, WQ, TERM
      COMMON /FPAR/ UF, VF, WF
C
C TEST FOR EQUALITY OF RHOI AND RHOJ
C
      EQUAL=RHOI.EQ.RHOJ
C
C SET DELTA PHI
C
      DELPHI=ATAN2(C4+C5,C6-C7-C8)
C
C COMMON SUB-EXPRESSIONS FOR THE FI AND FJ
C
      C45=C4+C5
      C68=C6-C8
      UN=0.5D0*C45*C45+C7*C7+C68*C68
      VN=2.D0*C7*C68-0.5D0*C45*C45
      WN=DSQRT(UN*UN-VN*VN)
C
C FI(DELPHI+PI)
C
      UF=RHOI*UN
      VF=RHOI*VN
      WF=RHOI*WN
      FIPLUS=FI(DELPHI+PI)
C
C FI(PI-DELPHI)
C
      FIMINU=FI(PI-DELPHI)
C
C FI(DELPHI-PI)
C
      FIUNIM=FI(DELPHI-PI)
```



```
C
C IF NEEDED, FI(-DELPHI)
C
      IF(DELPHI.GT.HALFPI) THEN
        FIMDP=FI(-DELPHI)
      END IF
C
C IF NEEDED FOR P1(J,I), COMPUTE INTEGRALS OF QJ*FI
C
      IF(.NOT.EQUAL) THEN
        UQ=RHOJ*UN
        VQ=RHOJ*VN
        WQ=RHOJ*WN
        TERM=1.D0+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
        CALL ADQUAD(PI-DELPHI, TWOPI-DELPHI, AJI, DGAU20, QF, 1.D-5,
     $              1.D-5, WORK, STACK, HEAP, 15, KODE)
        IF(KODE.NE.0) STOP 'P1(J,I;011) PARTA DID NOT CONVERGE'
        CALL ADQUAD(TWOPI-DELPHI,DELPHI+PI, BJI, DGAU20, QF, 1.D-5,
     $              1.D-5, WORK, STACK, HEAP, 15, KODE)
        IF(KODE.NE.0) STOP 'P1(J,I;011) PARTB DID NOT CONVERGE'
      END IF
C
C FJ(DELPHI+PI)
C
      UF=RHOJ*UN
      VF=RHOJ*VN
      WF=RHOJ*WN
      FJPLUS=FI(DELPHI+PI)
C
C FJ(DELPHI-PI)
C
      FJMINU=FI(DELPHI-PI)
C
C IF RHOI .NE. RHOJ, NEED FJ(PI-DELPHI) AND POSSIBLY FJ(-DELPHI)
C
      IF(.NOT.EQUAL) THEN
        FJUNIM=FI(PI-DELPHI)
        IF(DELPHI.GT.HALFPI) FJMDP=FI(-DELPHI)
      END IF
C
C THE INTEGRAL TERMS
C     (1) SET UP THE PARAMETERS FOR QI(X) FUNCTION
C
      UQ=UN*RHOI
      VQ=VN*RHOI
      WQ=WN*RHOI
      TERM=1.D0+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
C
C     (2) DO THE INTEGRATION FOR P1
C
      CALL ADQUAD(PI-DELPHI, TWOPI-DELPHI, PARTA, DGAU20, QF, 1.D-5,
     $            1.D-5, WORK, STACK, HEAP, 15, KODE)
```

```
      IF(KODE.NE.0) STOP 'P1(I,J|011) PARTA DID NOT CONVERGE'
      CALL ADQUAD(TWOPI-DELPHI, PI+DELPHI, PARTB, DGAU20, QF, 1.D-5,
     $     1.D-5, WORK, STACK, HEAP, 15, KODE)
      IF(KODE.NE.0) STOP 'P1(I,J|011) PARTB DID NOT CONVERGE'
C
C   (3) RETURN THE FUNCTION P1IJ
C
      P1IJ=FJPLUS*(FIPLUS-FIMINU)-PARTA-PARTB
      IF(DELPHI.GT.HALFPI) P1IJ=P1IJ+FJPLUS+FIPLUS-FIMDP
C
C   (4) DO THE INTEGRATION FOR P2
C
      CALL ADQUAD(DELPHI-PI, PI-DELPHI, RESULT, DGAU20, QF, 1.D-5,
     $     1.D-5, WORK, STACK, HEAP, 15, KODE)
      IF(KODE.NE.0) STOP 'P2(I,J|011) DID NOT CONVERGE'
C
C   (5) RETURN P2
C
      P2=RESULT-FJMINU*(FIMINU-FIUNIM)
      IF(HALFPI.GT.DELPHI) P2=P2+FIDMP-FIUNIM-FJMINU
C
C   (6) RETURN P1JI
C
      IF(EQUAL) THEN
        P1JI=P1IJ
      ELSE
        P1JI=FIPLUS*(FJPLUS-FJUNIM)-AJI-BJI
        IF(DELPHI.GT.HALFPI) P1JI=P1JI+FIPLUS+FJPLUS-FJMDP
      END IF
      RETURN
      END
```





```
      SUBROUTINE CP010(RHOI, RHOJ, P1IJ, P1JI, P2)
C
C COMPUTE CONDITIONAL PROBABILITIES:
C                                          .  P1(I,J|010)
C                                             P1(J,I|010)
C                                         AND P2(I,J|010)
C     NOTE: I=J IS TESTED TO AVOID UNNECESSARY RECOMPUTATION
C           OF P1JI
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (PI=3.14159265358979323846268D0)
      PARAMETER (HALFPI=1.57079632679489661923132D0)
      PAPAMETER (TWOPI=6.28318530717958647692536D0)
      DIMENSION WORK(15), STACK(15), HEAP(15)
      EXTERNAL DGAU20, QF
      LOGICAL EQUAL
      COMMON /CONST/ A0, A1, A2, A3, A4,
     $               C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $               C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
      COMMON /THINGS/ ARE, LELPHI, OMRSQ2
      COMMON /QPAR/ UQ, VQ, WQ, TERM
      COMMON /FPAR/ UF, VF, WF
C
C TEST FOR EQUALITY OF RHOI AND RHOJ
C
      EQUAL=RHOI.EQ.RHOJ
C
C SET DELTA PHI
C
      DELPHI=2.D0*ATAN2(C1,C2-C3)
C
C COMMON SUB-EXPRESSIONS FOR THE FI AND FJ
C
      C23=C2-C3
      UN=C1*C1+C23*C23
      VN=0.D0
      WN=UN
C
C FI(DELPHI+PI)
C
      UF=RHOI*UN
      VF=RHOI*VN
      WF=RHOI*WN
      FIPLUS=FI(DELPHI+PI)
C
C FI(PI-DELPHI)
C
      FIMINU=FI(PI-DELPHI)
C
C FI(DELPHI-PI)
C
      FIUNIM=FI(DELPHI-PI)
```

```
          C
          C IF NEEDED, FI(-DELPHI)
          C
0025            IF(DELPHI.GT.HALFPI) THEN
0026              FIMDP=FI(-DELPHI)
0027            END IF
          C
          C IF NEEDED FOR P1(J,I), COMPUTE INTEGRALS OF QJ*FI
          C
0028            IF(.NOT.EQUAL) THEN
0029              UQ=RHOJ*UN
0030              VQ=RHOJ*VN
0031              WQ=RHOJ*WN
0032              TERM=1.DO+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
0033              CALL ADQUAD(PI-DELPHI, TWOPI-DELPHI, AJI, DGAU20, QF, 1.D-5,
        $                     1.D-5, WORK, STACK, HEAP, 15, KODE)
0034              IF(KODE.NE.0) STOP 'P1(J,I!010) PARTA DID NOT CONVERGE'
0035              CALL ADQUAD(TWOPI-DELPHI,DELPHI+PI, BJI, DGAU20, QF, 1.D-5,
        $                     1.D-5, WORK, STACK, HEAP, 15, KODE)
0036              IF(KODE.NE.0) STOP 'P1(J,I!010) PARTB DID NOT CONVERGE'
0037            END IF
          C
          C FJ(DELPHI+PI)
          C
0038            UF=RHOJ*UN
0039            VF=RHOJ*VN
0040            WF=RHOJ*WN
0041            FJPLUS=FI(DELPHI+PI)
          C
          C FJ(DELPHI-PI)
          C
0042            FJMINU=FI(DELPHI-PI)
          C
          C IF RHOI .NE. RHOJ, NEED FJ(PI-DELPHI) AND POSSIBLY FJ(-DELPHI)
          C
0043            IF(.NOT.EQUAL) THEN
0044              FJUNIM=FI(PI-DELPHI)
0045              IF(DELPHI.GT.HALFPI) FJMDP=FI(-DELPHI)
0046            END IF
          C
          C THE INTEGRAL TERMS
          C     (1) SET UP THE PARAMETERS FOR QI(X) FUNCTION
          C
0047            UQ=UN*RHOI
0048            VQ=VN*RHOI
0049            WQ=WN*RHOI
0050            TERM=1.DO+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
          C
          C     (2) DO THE INTEGRATION FOR P1
          C
0051            CALL ADQUAD(PI-DELPHI, TWOPI-DELPHI, PARTA, DGAU20, QF, 1.D-5,
        $                   1.D-5, WORK, STACK, HEAP, 15, KODE)
```





```
0052            IF(KODE.NE.0) STOP 'P1(I,J!010) PARTA DID NOT CONVERGE'
0053            CALL ADQUAD(PI-DELPHI, PI+DELPHI, PARTB, DGAU20, QF, 1.D-5,
        $                   1.D-5, WORK, STACK, HEAP, 15, KODE)
0054            IF(KODE.NE.0) STOP 'P1(I,J!010) PARTB DID NOT CONVERGE'
          C
          C     (3) RETURN THE FUNCTION P1IJ
          C
0055            P1IJ=FJPLUS*(FIPLUS-FIMINU)-PARTA-PARTB
0056            IF(DELPHI.GT.HALFPI) P1IJ=P1IJ+FJPLUS+FIPLUS-FIMDP
          C
          C     (4) DO THE INTEGRATION FOR P2
          C
0057            CALL ADQUAD(DELPHI-PI, PI-DELPHI, RESULT, DGAU20, QF, 1.D-5,
        $                   1.D-5, WORK, STACK, HEAP, 15, KODE)
0058            IF(KODE.NE.0) STOP 'P2(I,J!010) DID NOT CONVERGE'
          C
          C     (5) RETURN P2
          C
0059            P2=RESULT-FJMINU*(FIMINU-FIUNIM)
0060            IF(HALFPI.GT.DELPHI) P2=P2+FIDMP-FIUNIM-FJMINU
          C
          C     (6) RETURN P1JI
          C
0061            IF(EQUAL) THEN
0062              P1JI=P1IJ
0063            ELSE
0064              P1JI=FIPLUS*(FJPLUS-FJUNIM)-AJI-BJI
0065              IF(DELPHI.GT.HALFPI) P1JI=P1JI+FIPLUS+FJPLUS-FJMDP
0066            END IF
0067            RETURN
0068            END
```

```
PDP-11 FORTRAN-77 V4.0-1        15:51:46     3-Dec-87              Page 21
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION QF(X)
        C INTEGRAND FUNCTION QI(X)*FJ(-X)
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            QF=QI(X)*FI(-X)
0004            RETURN
0005            END

PDP-11 FORTRAN-77 V4.0-1        15:51:47     3-Dec-87              Page 22
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION FI(PSI)
        C THE FUNCTION FI(PSI)
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETER (HALFPI = 1.570796326794896619231320D0)
0004            PARAMETER (HALMPI =-1.570796326794896619231320D0)
0005            PARAMETER (FOURPI =12.566370614359172953850570D0)
0006            EXTERNAL DGAUXX, FINNER
0007            DIMENSION WORK(15), STACK(15), HEAP(15)
0008            COMMON /THINGS/ ARE, DELPHI, OMRSQ2
0009            COMMON /FPAR/ UF, VF, WF
0010            COMMON /SIGH/ RCPSI, WCPMP, WSPMP, RSP, RCP
0011            RCPSI=ARE*COSINE(PSI)
0012            DIFF=DELPHI-PSI
0013            WCPMP=WF*COSINE(DIFF)
0014            WSPMP=WF*DSIN(DIFF)
0015            RSP=ARE*DSIN(PSI)
0016            RCP=ARE*COSINE(PSI)
0017            CALL ADQUA2(HALMPI, HALFPI, RESULT, DGAUXX, FINNER, 1.D-6,
          $                   1.D-6, WORK, STACK, HEAP, 15, KODE)
0018            IF(KODE.NE.0) THEN
0019              STOP 'FI(PSI) INTEGRAL FAILED'
0020            END IF
0021            FI=RESULT/FOURPI
0022            RETURN
0023            END

PDP-11 FORTRAN-77 V4.0-1        15:51:50     3-Dec-87              Page 23
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION FINNER(ALPHA)
        C THE INTEGRAND FUNCTION FOR FI(X)
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            COMMON /FPAR/ UF, VF, WF
0004            COMMON /SIGH/ RCPSI, WCPMP, WSPMP, RSP, RCP
0005            SA=DSIN(ALPHA)
0006            CA=COSINE(ALPHA)
0007            DDD=UF-VF*SA-WCPMP*CA
0008            BBB=1.D0-RCP*CA
0009            FINNER=DEXP(-DDD/BBB)*(WSPMP/DDD+RSP/BBB)
0010            RETURN
0011            END

PDP-11 FORTRAN-77 V4.0-1        15:51:52     3-Dec-87              Page 24
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION QI(X)
        C
        C THE FUNCTION QI(X)
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            EXTERNAL DGAUXX, QINNER
0004            PARAMETER (HALFPI = 1.570796326794896619231320D0)
0005            PARAMETER (HALMPI =-1.570796326794896619231320D0)
0006            PARAMETER (TWOPI=6.283185307179586476925360D0)
0007            DIMENSION WORK(15), STACK(15), HEAP(15)
0008            COMMON /THINGS/ ARE, DELPHI, OMRSQ2
0009            COMMON /QPAR/ UQ, VQ, WQ, TERM
0010            COMMON /ECKS/ WCPMX, RCX
0011            WCPMX=WQ*COSINE(DELPHI-X)
0012            RCX=ARE*COSINE(X)
0013            CALL ADQUA2(HALMPI, HALFPI, RESULT, DGAUXX, QINNER, 1.D-6,
          $                   1.D-6, WORK, STACK, HEAP, 15, KODE)
0014            IF(KODE.NE.0) STOP 'QI INTEGRAL FAILED'
0015            QI=OMRSQ2*RESULT/TWOPI
0016            RETURN
0017            END

PDP-11 FORTRAN-77 V4.0-1        15:51:56     3-Dec-87              Page 25
CPFSK2.FTN;32            /F77/TR:BLOCKS/WR

0001            DOUBLE PRECISION FUNCTION QINNER(U)
        C
        C THE INTEGRAND FUNCTION FOR QI
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            COMMON /QPAR/ UQ, VQ, WQ, TERM
0004            COMMON /ECKS/ WCPMX, RCX
0005            CU=COSINE(U)
0006            SU=DSIN(U)
0007            AAA=(UQ-VQ*SU-WCPMX*CU)/(1.D0-RCX*CU)
0008            QINNER=CU*DEXP(-AAA)*(TERM-AAA)
0009            RETURN
0010            END
```

```
      SUBROUTINE ADQUAD(XL,XU,Y,QR,F,TOL,ABSTOL,
     $                  WORK,STACK,HEAP,N,KODE)
C
C ADAPTIVE QUADRATURE ALGORITHM FOR NUMERICAL INTEGRATION
C
C     XL - LOWER LIMIT OF INTEGRAL (IN)
C     XU - UPPER LIMIT OF INTEGRAL (IN)
C     Y - VALUE OF INTEGRAL (OUT)
C     QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C               WITH CALLING SEQUENCE
C               CALL QR(XL,XU,F,Y)
C      F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C    TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C ABSTOL - ABSOLUTE ERROR TOLERANCE
C   WORK - WORK ARRAY OF SIZE N (IN)
C  STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
C   HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
C          SAME ARRAY AS WORK (IN)
C      N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C   KODE - ERROR INDICATOR (OUT)
C          0 -- NO ERROR
C          1 -- WORK ARRAYS TOO SMALL
C
C R. H. FRENCH, 14 AUGUST 1984
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      EXTERNAL F
      DIMENSION WORK(N),STACK(N),HEAP(N)
      KODE=0
      Y=0.D0
      WORK(1)=XU
      CALL QR(XL,XU,F,T)
      HEAP(1)=T
      A=XL
      NPTS=1
      EPS=TOL
      STACK(1)=EPS
10    B=WORK(NPTS)
      XM=(A+B)*0.5D0
      CALL QR(A,XM,F,P1)
      CALL QR(XM,B,F,P2)
      TEST=DMAX1(EPS*DABS(T),ABSTOL)
      IF(DABS(T-P1-P2).LE.TEST .OR. DABS(T).LE.ABSTOL) GOTO 20
C SPLIT IT
      NPTS=NPTS+1
      IF(NPTS.GT.N) THEN
        Y=P1+P2
        KODE=1
        RETURN
      END IF
      WORK(NPTS)=XM
      HEAP(NPTS)=P2
```



```
      T=P1
      EPS=DMAX1(EPS/2.D0,5.D-16)
      STACK(NPTS)=EPS
      GOTO 10
C FINISHED A PIECE
20    Y=Y+P1+P2
      EPS=STACK(NPTS)
      T=HEAP(NPTS)
      NPTS=NPTS-1
      A=B
      IF(NPTS.EQ.0) RETURN
      GOTO 10
      END
```

```
      SUBROUTINE ADQUA2(XL,XU,Y,QR,F,TOL,ABSTOL,
     $                  WORK,STACK,HEAP,N,KODE)
C
C ADAPTIVE QUADRATURE ALGORITHM FOR NUMERICAL INTEGRATION
C
C     XL - LOWER LIMIT OF INTEGRAL (IN)
C     XU - UPPER LIMIT OF INTEGRAL (IN)
C     Y - VALUE OF INTEGRAL (OUT)
C     QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
C              WITH CALLING SEQUENCE
C              CALL QR(XL,XU,F,Y)
C      F - NAME OF FUNCTION TO BE INTEGRATED (IN)
C    TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
C ABSTOL - ABSOLUTE ERROR TOLERANCE
C   WORK - WORK ARRAY OF SIZE N (IN)
C  STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
C   HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
C          SAME ARRAY AS WORK (IN)
C      N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
C   KODE - ERROR INDICATOR (OUT)
C          0 -- NO ERROR
C          1 -- WORK ARRAYS TOO SMALL
C
C R. H. FRENCH, 14 AUGUST 1984
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      EXTERNAL F
      DIMENSION WORK(N),STACK(N),HEAP(N)
      KODE=0
      Y=0.D0
      WORK(1)=XU
      CALL QR(XL,XU,F,T)
      HEAP(1)=T
      A=XL
      NPTS=1
      EPS=TOL
      STACK(1)=EPS
 10   B=WORK(NPTS)
      XM=(A+B)*0.5D0
      CALL QR(A,XM,F,P1)
      CALL QR(XM,B,F,P2)
      TEST=DMAX1(EPS*DABS(T),ABSTOL)
      IF(DABS(T-P1-P2).LE.TEST .OR. DABS(T).LE.ABSTOL) GOTO 20
C SPLIT IT
      NPTS=NPTS+1
      IF(NPTS.GT.N) THEN
        Y=P1+P2
        KODE=1
        RETURN
      END IF
      WORK(NPTS)=XM
      HEAP(NPTS)=P2
```



```
      T=P1
      EPS=DMAX1(EPS/2.D0,5.D-16)
      STACK(NPTS)=EPS
      GOTO 10
C FINISHED A PIECE
 20   Y=Y+P1+P2
      EPS=STACK(NPTS)
      T=HEAP(NPTS)
      NPTS=NPTS-1
      A=B
      IF(NPTS.EQ.0) RETURN
      GOTO 10
      END
```

```
PDP-11 FORTRAN-77 V4.0-1        15:52:06     3-Dec-87              Page 30
CPFSK2.FTN;32              /F77/TR:BLOCKS/WR

0001            SUBROUTINE DGAUXX(A,B,F,ANSWER)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                                                        C
          C 20-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL C
          C                                                        C
          C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4 C
          C                                                        C
          C R. H. FRENCH, 21 JUNE 1983                             C
          C                                                        C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            DIMENSION X(10),W(10)
0004            DATA X/ 0.07652652113349733375500,
              1         0.22778585114164507808000,
              1         0.37370608871541956067300,
              1         0.51086700195082709800400,
              1         0.63605368072651502545300,
              1         0.74633190646015079261400,
              1         0.83911697182221882339500,
              1         0.91223442825132590586800,
              1         0.96397192727791379126800,
              1         0.99312859918509492478600 /
0005            DATA W/ 0.15275338713072585069800,
              1         0.14917298647260374678800,
              1         0.14209610931838205132900,
              1         0.13168863844917662689800,
              1         0.11819453196151841731200,
              1         0.10193011981724043503700,
              1         0.08327674157670474872500,
              1         0.06267204833410906357000,
              1         0.04060142980038694133100,
              1         0.01761400713915211831200 /
0006            ANSWER=0.D0
0007            BMAO2=(B-A)/2.D0
0008            BPAO2=(B+A)/2.D0
0009            DO 10 I=1,10
0010            C=X(I)*BMAO2
0011            Y1=BPAO2+C
0012            Y2=BPAO2-C
0013            ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0014      10    CONTINUE
0015            ANSWER=ANSWER*BMAO2
0016            RETURN
0017            END
```

```
PDP-11 FORTRAN-77 V4.0-1        15:52:09     3-Dec-87              Page 31
CPFSK2.FTN;32              /F77/TR:BLOCKS/WR

0001            SUBROUTINE DGAU20(A,B,F,ANSWER)
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          C                                                        C
          C 20-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL C
          C                                                        C
          C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4 C
          C                                                        C
          C R. H. FRENCH, 21 JUNE 1983                             C
          C                                                        C
          CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            DIMENSION X(10),W(10)
0004            DATA X/ 0.07652652113349733375500,
              1         0.22778585114164507808000,
              1         0.37370608871541956067300,
              1         0.51086700195082709800400,
              1         0.63605368072651502545300,
              1         0.74633190646015079261400,
              1         0.83911697182221882339500,
              1         0.91223442825132590586800,
              1         0.96397192727791379126800,
              1         0.99312859918509492478600 /
0005            DATA W/ 0.15275338713072585069800,
              1         0.14917298647260374678800,
              1         0.14209610931838205132900,
              1         0.13168863844917662689800,
              1         0.11819453196151841731200,
              1         0.10193011981724043503700,
              1         0.08327674157670474872500,
              1         0.06267204833410906357000,
              1         0.04060142980038694133100,
              1         0.01761400713915211831200 /
0006            ANSWER=0.D0
0007            BMAO2=(B-A)/2.D0
0008            BPAO2=(B+A)/2.D0
0009            DO 10 I=1,10
0010            C=X(I)*BMAO2
0011            Y1=BPAO2+C
0012            Y2=BPAO2-C
0013            ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0014      10    CONTINUE
0015            ANSWER=ANSWER*BMAO2
0016            RETURN
0017            END
```

```
0001            DOUBLE PRECISION FUNCTION COSINE(X)
        C
        C INSURE DCOS(0.D0) = 1.0000000000000000D0 EXACTLY
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            IF(X.EQ.0.D0) THEN
0004              COSINE=1.D0
0005            ELSE
0006              COSINE=DCOS(X)
0007            END IF
0008            RETURN
0009            END
```

# APPENDIX D

## A PROGRAM FOR THE CPFSK ERROR PROBABILITY USING THE DFT METHOD

```
        PROGRAM FSKFFT
C
C PROBABILITY OF ERROR FOR FH/CPFSK USING L-FOLD DIVERSITY
C COMPUTED BY USING FFT TO GET CHARACTERISTIC FUNCTION,
C AVERAGING CHARACTERISTIC FUNCTION OVER PATTERNS, AND
C FINALLY TAKING INVERSE FFT OF L-TH POWER OF CHARACTERISTIC
C FUNCTION TO OBTAIN THE P.D.F. WHICH IS INTEGRATED NUMERICALLY.
C
C J.S. LEE ASSOCIATES, INC.
C 2001 JEFFERSON DAVIS HWY., SUITE 601
C ARLINGTON, VIRGINIA 22202
C
C ANALYSIS: L.E. MILLER                PROGRAM: R.H. FRENCH
C 9 DEC 1987
C
        IMPLICIT DOUBLE PRECISION(A-H,O-Z)
        PARAMETER (IDIM=8192)
        PARAMETER (PI=3.14159265358979323846D0)
        REAL*4 TIME, SECNDS
        VIRTUAL Z(2,IDIM), ZAVG(2,IDIM)
        COMMON /PARMS/ EBNODB, EBNJDB,GAMMA, L, H, D
        COMMON /ROSE/ RHON, RHOT
        COMMON /SIZES/ ISIZE, JSIZE, LSIZE, ISIZE2
        COMMON /CONST/ AO, A1, A2, A3, A4,
     $                 C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $                 C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
        COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
        CALL JSLGO
        CALL UFLOFF
        DX=PI/128.DO
C
C PARAMETERS FOR THIS RUN
C
        CALL GET
        TIME=SECNDS(0.)
C
C COMPUTE CONSTANTS
C
        PIH=PI*H
        ARE=DEXP(-PI*D*D)
        OMRSQ2=(1.DO-ARE*ARE)/2.DO
        AO=EXP(-PIH*H/(8.DO*D*D))
        A1=EXP(-PI/(8.DO*D*D))
        A2=EXP(-PI/(2.DO*D*D))
        A3=EXP(-PI/(32.DO*D*D))
        A4=EXP(-9.DO*PI/(32.DO*D*D))
        C1=(4.DO*H/PI)*DCOS(PIH/2.DO)*A1/(1.DO-H*H)
        C1PM=-C1*PI
        C2=DSINC(H/2.DO)
        C3=(4.DO*H/PI)*DSIN(PIH/2.DO)*A2/(4.DO-H*H)
        C3P2=2.DO*PI*C3
        DENO=1.DO-4.DO*H*H
```



```
        IF(DENO.NE.0.DO) THEN
          C4=(8.DO*H/PI)*DCOS(PIH)*A3/DENO
        ELSE
C L'HOSPITAL'S RULE WHEN THE DENOMINATOR VANISHES (H=1/2)
          C4=A3
        END IF
        C4PH=C4*PI/2.DO
        C5=(8.DO*H/PI)*DCOS(PIH)*A4/(9.DO-4.DO*H*H)
        C5P3R2=1.5DO*C5*PI
        C6=DSINC(H)
        C7=C6*2 DO*H*H*A1/(1.DO-H*H)
        C7P=C7*PI
        C8=C6*2.DO*H*H*A2/(4.DO-H*H)
        C8P2=C8*PI*2.DO
        DL=D*L
        EBNO=10.DO**(EBNODB/10.DO)
        RHON=EBNO/DL
        EBNJ=10.DO**(EBNJDB/10.DO)
        GEBNJ=GAMMA*EBNJ
        EBNT=GEBNJ*EBNO/(GEBNJ+EBNO)
        RHOT=EBNT/DL
        WRITE(6,1) EBNODB, EBNJDB, H, D, L, GAMMA
    1   FORMAT(' FH/CPFSK USING FFT; L-FOLD DIVERSITY'/
     $         ' SIMPSON''S RULE INTEGRATION'//
     $         ' EB/NO = ',F7.3,' dB',5X,'EB/NJ = ',F6.2,' dB'/
     $         ' H = ',F5.3,5X,'D = ',F4.2/
     $         ' L = ',I1/' GAMMA = ',1PD11.4)
C
C CREATE THE PDF
C
        CALL MAKEFL(Z, ZAVG, NRIGHT)
C
C INTEGRATE TO GET P(E)
C
        CALL PSUBE(ZAVG, NRIGHT, PE)
        WRITE(6,70) PE
   70   FORMAT(//' P(E) = ',1PD12.5)
        TIME=SECNDS(TIME)
        WRITE(6,999) TIME
  999   FORMAT(////' ELAPSED TIME = ',F7.2,' SECONDS')
        STOP
        END
```

```
          SUBROUTINE GET
C
C PARAMETER INPUT
C
          IMPLICIT DOUBLE PRECISION(A-H,O-Z)
          PARAMETER (PI=3.14159265358979323846D0)
          CHARACTER*6 FIELD, BLANK6
          DIMENSION LSZ(4)
          COMMON /PARMS/ EBNODB, EBNJDB, GAMMA, L, H, D
          COMMON /SIZES/ ISIZE, JSIZE, LSIZE, ISIZE2
          DATA BLANK6/'      '/
          DATA LSZ/8, 11, 12, 13/
    10    WRITE(5,11)
    11    FORMAT(' ENTER EB/NO IN DECIBELS [11.257 dB]: ',$)
          READ(5,12,ERR=10) FIELD
    12    FORMAT(A6)
          IF(FIELD.EQ.BLANK6) THEN
            EBNODB=11.257
          ELSE
            READ(FIELD,13,ERR=10) EBNODB
    13      FORMAT(BN,F6.0)
          END IF
    20    WRITE(5,21)
    21    FORMAT(' ENTER EB/NJ IN DECIBELS [20 dB]: ',$)
          READ(5,22) FIELD
    22    FORMAT(A6)
          IF(FIELD.EQ.BLANK6) THEN
            EBNJDB=20.D0
          ELSE
            READ(FIELD,23,ERR=20) EBNJDB
    23      FORMAT(BN,F6.0)
          END IF
    30    WRITE(5,31)
    31    FORMAT(' ENTER H [0.7D0]: ',$)
          READ(5,32,ERR=30) H
    32    FORMAT(BN,F10.0)
          IF(H.EQ.0.D0) H=0.7D0
          IF(H.LT.0.D0) GOTO 30
    40    WRITE(5,41)
    41    FORMAT(' ENTER D [1.0]: ',$)
          READ(5,42,ERR=40) D
    42    FORMAT(BN,F10.0)
          IF(D.EQ.0.D0) D=1.0D0
          IF(D.LT.0.D0 .OR. D.GT.1.D0) GOTO 40
    50    WRITE(5,51)
    51    FORMAT(' ENTER GAMMA [0.001]: ',$)
          READ(5,52,ERR=50) GAMMA
    52    FORMAT(BN,F10.0)
          IF(GAMMA.EQ.0.D0) GAMMA=0.001D0
          IF(GAMMA.LT.0.D0 .OR. GAMMA.GT.1.D0) GOTO 50
    60    WRITE(5,61)
    61    FORMAT(' ENTER L [3]: ',$)
```



```
          READ(5,62,ERR=60) L
    62    FORMAT(I1)
          IF(L.EQ.0) L=3
          IF(L.LT.0 .OR. L.GT.4) GOTO 60
          LSIZE=LSZ(L)
          ISIZE=2**LSIZE
          ISIZE2=ISIZE/4+1
          JSIZE=256
          RETURN
          END
```

```
0001            SUBROUTINE MAKEFL(Z, ZAVG, NRIGHT)
          C
          C CREATE THE PDF FOR L-FOLD DIVERSITY SUM
          C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETFR (IDIM=8192)
0004            PARAMETER (IDIM2=IDIM/4+1)
0005            PARAMETER    (PI=3.14159265358979323846D0)
0006            PARAMETER (TWOPI=6.28318530717958647692D0)
0007            CHARACTER*13 FTRIG
0008            VIRTUAL Z(2,IDIM), ZAVG(2,IDIM), COEF(IDIM2)
0009            LOGICAL VALID, DISK
0010            EXTERNAL BCI111, BCI011, BCI010
0011            DIMENSION CU111(4),  CU011(4),  CU010(4),
             $            CJ111(4),  CJ011(4),  CJ010(4),
             $            PU111(256),PU011(256),PU010(256),
             $            PJ111(256),PJ011(256),PJ010(256)
0012            COMMON /PARMS/ EBNODB, EBNJDB, GAMMA, L, H, D
0013            COMMON /ROSE/ RHON, RHOT
0014            COMMON /SIZES/ ISIZE, JSIZE, LSIZE, ISIZE2
0015            COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
0016            LTS=ISIZE/4+1
0017            WRITE(FTRIG,400) ISIZE
0018        400 FORMAT('TRIG',I5.5,'.DAT')
0019            OPEN(UNIT=1,FILE=FTRIG,STATUS='OLD',FORM='UNFORMATTED',
             $       ACCESS='SEQUENTIAL',ERR=450,READONLY)
          C IF OPEN IS SUCCESSFUL, WE HAVE A TRIG TABLE ON DISK
0020            READ(1,ERR=410) (COEF(I),I=1,LTS)
0021            CLOSE(UNIT=1)
0022            DISK=.TRUE.
0023            VALID=.TRUE.
0024            GOTO 499
0025        410 STOP 'ERROR READING TRIG TABLE FILE'
          C WE HAVE NO FILE, SO SET FLAG TO MAKE ONE
0026        450 DISK=.FALSE.
0027            VALID=.FALSE.
          C
          C CLEAR THE AVERAGED FFT ARRAY
          C
0028        499 DO 1 I=1,ISIZE
0029            ZAVG(1,I)=0.D0
0030            ZAVG(2,I)=0.D0
0031          1 CONTINUE
0032            OMG=1.D0-GAMMA
          C
          C COMPUTE AVERAGE CLICK NUMBERS AND CLICK PROBABILITIES
          C
          C  — UNJAMMED CASES
0033            CALL BARN(RHON, BCI111, BNC111)
0034            CALL CLICKS(BNC111, CU111, L)
0035            CALL BARN(RHON, BCI011, BNC011)
0036            CALL CLICKS(BNC011, CU011, L)
```





```
0037            CALL BARN(RHON, BCI010, BNC010)
0038            CALL CLICKS(BNC010, CU010, L)
          C  -- JAMMED CASES
0039            CALL BARN(RHOT, BCI111, BNC111)
0040            CALL CLICKS(BNC111, CJ111, L)
0041            CALL BARN(RHOT, BCI011, BNC011)
0042            CALL CLICKS(BNC011, CJ011, L)
0043            CALL BARN(RHOT, BCI010, BNC010)
0044            CALL CLICKS(BNC010, CJ010, L)
          C
          C COMPUTE DENSITY FUNCTIONS MODULO TWO PI
          C
          C — UNJAMMED CASES
0045            CALL CP111(RHON)
0046            NPU111=NPHI
0047            CALL MAKPDF(PU111,NOU111)
0048            CALL CP011(RHON)
0049            NPU011=NPHI
0050            CALL MAKPDF(PU011,NOU011)
0051            CALL CP010(RHON)
0052            NPU010=NPHI
0053            CALL MAKPDF(PU010,NOU010)
          C — JAMMED CASES
0054            CALL CP111(RHOT)
0055            NPJ111=NPHI
0056            CALL MAKPDF(PJ111,NOJ111)
0057            CALL CP011(RHOT)
0058            NPJ011=NPHI
0059            CALL MAKPDF(PJ011,NOJ011)
0060            CALL CP010(RHOT)
0061            NPJ010=NPHI
0062            CALL MAKPDF(PJ010,NOJ010)
0063            K=L-1
          C
          C -------------------------------------------------------------------
          C                              PATTERN 111 PDF
          C -------------------------------------------------------------------
          C
          C
          C     CLEAR THE FFT ARRAY IN PREPARATION FOR ASSEMBLING THE INPUT
          C
0064            DO 2 I=1,ISIZE
0065            Z(1,I)=0.D0
0066            Z(2,I)=0.D0
0067          2 CONTINUE
0068            CALL BUILD(CU111, CJ111, PU111, NOU111, PJ111, NOJ111,
             $             Z, K, NR111, CS111)
          C
          C TAKE FFT OF THE DENSITY FUNCTION
          C
0069            CALL DVFFTS(ISIZE, ISIZE2, Z, 1.D0, COEF, VALID)
          C
```

```
C
C AFTER FIRST CALL TO DVFFTS, WE MAY WANT TO SAVE A TRIG TABLE FILE
C ON DISK IF IT DOES NOT ALREADY EXIST
C
      IF(.NOT.DISK .AND. VALID) THEN
        OPEN(UNIT=1,FILE=FTRIG,STATUS='NEW',ACCESS='SEQUENTIAL',
     $       FORM='UNFORMATTED')
        WRITE(1) (COEF(IT),IT=1,LTS)
        CLOSE(UNIT=1)
C
C TOGGLE THE DISK FLAG SO WE DON'T DUPLICATE THE FILE
C
        DISK=.TRUE.
      END IF
C
C NORMALIZE AND RAISE THE CHARACTERISTIC FUNCTION TO THE L-TH POWER
C
      PNORM=CS111/(Z(1,1)*DX)
      WRITE(6,18) PNORM
   18 FORMAT(' PATTERN 111 NORMALIZATION FACTOR =',1PD11.4)
      DO 20 I=1,ISIZE
      Z1=Z(1,I)*PNORM
      W1=Z1
      Z2=Z(2,I)*PNORM
      W2=Z2
      DO 19 J=1,K
      TEMP=Z1*W1-Z2*W2
      Z2=Z1*W2+Z2*W1
      Z1=TEMP
   19 CONTINUE
C MOVE WEIGHTED RESULT INTO AVERAGE FFT ARRAY
      ZAVG(1,I)=Z1*0.25D0
      ZAVG(2,I)=Z2*0.25D0
   20 CONTINUE
C
C -----------------------------------------------------------------
C                          PATTERN 011 PDF
C -----------------------------------------------------------------
C
C
C     CLEAR THE FFT ARRAY IN PREPARATION FOR ASSEMBLING THE INPUT
C
      DO 22 I=1,ISIZE
      Z(1,I)=0.D0
      Z(2,I)=0.D0
   22 CONTINUE
      CALL BUILD(CU011, CJ011, PU011, NOU011, PJ011, NOJ011,
     $           Z, K, NR011, CS011)
C
C TAKE FFT OF THE DENSITY FUNCTION
C
      CALL DVFFTS(ISIZE, ISIZE2, Z, 1.D0, COEF, VALID)
```



```
C
C NORMALIZE AND RAISE THE CHARACTERISTIC FUNCTION TO THE L-TH POWER
C
      PNORM=CS011/(Z(1,1)*DX)
      WRITE(6,230) PNORM
  230 FORMAT(' PATTERN 011 NORMALIZATION FACTOR =',1PD11.4)
      DO 24 I=1,ISIZE
      Z1=Z(1,I)*PNORM
      W1=Z1
      Z2=Z(2,I)*PNORM
      W2=Z2
      DO 23 J=1,K
      TEMP=Z1*W1-Z2*W2
      Z2=Z1*W2+Z2*W1
      Z1=TEMP
   23 CONTINUE
      ZAVG(1,I)=Z1*0.5D0+ZAVG(1,I)
      ZAVG(2,I)=Z2*0.5D0+ZAVG(2,I)
   24 CONTINUE
C
C -----------------------------------------------------------------
C                          PATTERN 010 PDF
C -----------------------------------------------------------------
C
C     CLEAR THE FFT ARRAY IN PREPARATION FOR ASSEMBLING THE INPUT
C
      DO 25 I=1,ISIZE
      Z(1,I)=0.D0
      Z(2,I)=0.D0
   25 CONTINUE
      CALL BUILD(CU010, CJ010, PU010, NOU010, PJ010, NOJ010,
     $           Z, K, NR010, CS010)
C
C TAKE FFT OF THE DENSITY FUNCTION
C
      CALL DVFFTS(ISIZE, ISIZE2, Z, 1.D0, COEF, VALID)
C
C NORMALIZE AND RAISE THE CHARACTERISTIC FUNCTION TO THE L-TH POWER
C
      PNORM=CS010/(Z(1,1)*DX)
      WRITE(6,251) PNORM
  251 FORMAT(' PATTERN 010 NORMALIZATION FACTOR =',1PD11.4)
      DO 27 I=1,ISIZE
      Z1=Z(1,I)*PNORM
      W1=Z1
      Z2=Z(2,I)*PNORM
      W2=Z2
      DO 26 J=1,K
      TEMP=Z1*W1-Z2*W2
      Z2=Z1*W2+Z2*W1
      Z1=TEMP
   26 CONTINUE
```

```
PDP-11 FORTRAN-77 V4.0-1          07:40:57     14-Dec-87             Page 9
         C ADD WEIGHTED RESULT INTO AVERAGE FFT ARRAY
0133              ZAVG(1,I)=Z1*0.25D0+ZAVG(1,I)
0134              ZAVG(2,I)=Z2*0.25D0+ZAVG(2,I)
0135        27    CONTINUE
0136              NRIGHT=MAX0(NR111, NR011, NR010)
         C
         C -------------------------------------------------------------
         C      INVERSE FFT OF CHARACTERISTIC FUNCTION AVERAGED OVER PATTERNS
         C -------------------------------------------------------------
         C
0137              CALL DISFFT(ZAVG, ISIZE, ISIZE2, LSIZE, COEF)
0138              RETURN
0139              END
```

```
PDP-11 FORTRAN-77 V4.0-1          07:41:19     14-Dec-87             Page 10
CPFSKFFT.FTN;23            /F77/TR:BLOCKS/WR

0001              SUBROUTINE BARN(ROW, FNBAR, RESULT)
         C
         C COMPUTE ABSOLUTE VALUE OF AVERAGE CLICK NUMBER, BY
         C NUMERICAL INTEGRATION OF THE SPECIFIED FUNCTION FNBAR
         C
0002              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003              PARAMETER (TWOPI=6.2831853071795864769253600)
0004              DIMENSION WORK(15), STACK(15), HEAP(15)
0005              EXTERNAL FNBAR, DGAU20
0006              COMMON /ROZE/ RHO
0007              RHO=ROW
0008              CALL ADQUAD(-1.D0, 0.D0, BARNC, DGAU20, FNBAR, 1.D-5,
         $                  1.D-5, WORK, STACK, HEAP, 15, KODE)
0009              IF(KODE.NE.0) STOP 'CLICK TERM FAILED TO CONVERGE'
0010              RESULT=DABS(BARNC/TWOPI)
0011              RETURN
0012              END
```

```
PDP-11 FORTRAN-77 V4.0-1          07:41:21     14-Dec-87             Page 11
CPFSKFFT.FTN;23            /F77/TR:BLOCKS/WR

0001              DOUBLE PRECISION FUNCTION BCI111(X)
         C
         C INTEGRAND FUNCTION FOR CLICK NUMBER FOR PATTERN 111
         C
0002              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003              COMMON /CONST/ A0, A1, A2, A3, A4,
         $                  C1, C2, C3, C4, C5, C6, C7, C8, PIH,
         $                  C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
0004              COMMON /ROZE/ RHO
0005              PHX=PIH*X
0006              UPRIME=A0*DSIN(PHX)
0007              VPRIME=A0*COSINE(PHX)
0008              W=PIH*VPRIME
0009              Z=-PIH*UPRIME
0010              DDD=UPRIME*UPRIME + VPRIME*VPRIME
0011              BCI111 = DEXP(-RHO*DDD) * (W*VPRIME-UPRIME*Z)/DDD
0012              RETURN
0013              END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:41:23    14-Dec-87          Page 12
CPFSKFFT.FTN;23         /F77/TR:BLOCKS/WR

            DOUBLE PRECISION FUNCTION BCI011(X)
      C
      C INTEGRAND FUNCTION FOR CLICK NUMBER FOR PATTERN 011
      C
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            PARAMETER (PI=3.14159265358979323846268D0)
            PARAMETER (TWOPI=6.28318530717958647692536D0)
            PARAMETER (HALFPI=1.57079632679489661923132D0)
            PARAMETER (PI1R5=4.71238898038468985769396D0)
            COMMON /CONST/ A0, A1, A2, A3, A4,
     $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $                     C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
            COMMON /ROZE/ RHO
            HPX=HALFPI*X
            P15X=PI1R5*X
            TPX=TWOPI*X
            PIX=PI*X
            UPRIME=C4*DSIN(HPX) - C5*DSIN(P15X)
            VPRIME=C6 + C7*COSINE(PIX) - C8*COSINE(TPX)
            W=C4PH*COSINE(HPX) - C5P3R2*COSINE(P15X)
            Z=C8P2*DSIN(TPX) - C7P*DSIN(PIX)
            DDD=UPRIME*UPRIME + VPRIME*VPRIME
            BCI011 = DEXP(-RHO*DDD) * (W*VPRIME-UPRIME*Z)/DDD
            RETURN
            END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:41:27    14-Dec-87          Page 13
CPFSKFFT.FTN;23         /F77/TR:BLOCKS/WR

            DOUBLE PRECISION FUNCTION BCI010(X)
      C
      C INTEGRAND FUNCTION FOR CLICK NUMBER FOR PATTERN 010
      C
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            PARAMETER (PI=3.14159265358979323846268D0)
            PARAMETER (TWOPI=6.28318530717958647692536D0)
            COMMON /CONST/ A0, A1, A2, A3, A4,
     $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $                     C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
            COMMON /ROZE/ RHO
            TPX=TWOPI*X
            PIX=PI*X
            UPRIME=C1*COSINE(PIX)
            VPRIME=C2-C3*COSINE(TPX)
            W=C1PM*DSIN(PIX)
            Z=C3P2*DSIN(TPX)
            DDD=UPRIME*UPRIME + VPRIME*VPRIME
            BCI010 = DEXP(-RHO*DDD) * (W*VPRIME-UPRIME*Z)/DDD
            RETURN
            END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:41:30    14-Dec-87          Page 14
CPFSKFFT.FTN;23         /F77/TR:BLOCKS/WR

            SUBROUTINE CLICKS(BN, PIK, L)
      C
      C COMPUTE ARRAY OF CLICK PROBABILITIES, GIVEN AVERAGE CLICK NUMBER
      C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            DIMENSION PIK(4)
            DN=DABS(BN)
            PIK(1)=DEXP(-DN)
            IF(L.EQ.1) RETURN
            DO 10 K=1,L-1
            PIK(K+1)=PIK(K)*DN/K
         10 CONTINUE
            RETURN
            END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:41:32    14-Dec-87          Page 15
CPFSKFFT.FTN;23         /F77/TR:BLOCKS/WR

            SUBROUTINE CP111(RHO)
      C
      C SETUP FOR COMPUTING PDF FOR PATTERN 111
      C
            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
            COMMON /CONST/ A0, A1, A2, A3, A4,
     $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH,
     $                     C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
            COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
            COMMON /QPAR/ UQ, VQ, WQ, TERM
            DELPHI=PIH
      C DELPHI ROUNDED TO NEAREST MULTIPLE OF DX
            NPHI=DELPHI/DX+0.5D0
            IF(NPHI.GT.256) NPHI=256
            IF(NPHI.LE.0) NPHI=1
            DELPHI=DX*NPHI
            UN=A0*A0
            VN=0.D0
            WN=UN
            UQ=RHO*UN
            VQ=RHO*VN
            WQ=RHO*WN
            TERM=1.D0+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
            RETURN
            END
```

```
              SUBROUTINE CP011(RHO)
        C
        C SETUP FOR COMPUTING PDF FOR PATTERN 011
        C
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              COMMON /CONST/ A0, A1, A2, A3, A4,
           $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH,
           $                     C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
              COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
              COMMON /QPAR/ UQ, VQ, WQ, TERM
              DELPHI=ATAN2(C4+C5,C6-C7-C8)
        C DELPHI ROUNDED TO NEAREST MULTIPLE OF DX
              NPHI=DELPHI/DX+0.5D0
              IF(NPHI.GT.256) NPHI=256
              IF(NPHI.LE.0) NPHI=1
              DELPHI=DX*NPHI
              C45=C4+C5
              C68=C6-C8
              UN=0.5D0*C45*C45+C7*C7+C68*C68
              VN=2.D0*C7*C68-0.5D0*C45*C45
              WN=DSQRT(UN*UN-VN*VN)
              UQ=RHO*UN
              VQ=RHO*VN
              WQ=RHO*WN
              TERM=1.D0+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
              RETURN
              END
```





```
              SUBROUTINE CP010(RHO)
        C
        C SETUP FOR COMPUTING PDF FOR PATTERN 010
        C
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              COMMON /CONST/ A0, A1, A2, A3, A4,
           $                     C1, C2, C3, C4, C5, C6, C7, C8, PIH,
           $                     C1PM, C3P2, C4PH, C5P3R2, C7P, C8P2
              COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
              COMMON /QPAR/ UQ, VQ, WQ, TERM
              DELPHI=2.D0*ATAN2(C1,C2-C3)
        C DELPHI ROUNDED TO NEAREST MULTIPLE OF DX
              NPHI=DELPHI/DX+0.5D0
              IF(NPHI.GT.256) NPHI=256
              IF(NPHI.LE.0) NPHI=1
              DELPHI=DX*NPHI
              C23=C2-C3
              UN=C1*C1+C23*C23
              VN=0.D0
              WN=UN
              UQ=RHO*UN
              VQ=RHO*VN
              WQ=RHO*WN
              TERM=1.D0+(UQ-WQ*ARE*COSINE(DELPHI))/OMRSQ2
              RETURN
              END
```



```
              SUBROUTINE MAKPDF(PARRAY,NZERO)
        C
        C FILL AN ARRAY WITH VALUES OF THE PDF AT INCREMENTS OF PI/128
        C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              PARAMETER (PI=3.14159265358979323846D0)
              PARAMETER (NPI=128)
              DIMENSION PARRAY(256)
              COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
              DO 100 I=1,256
              J=I+NPHI-NPI
              IF(J.EQ.0) NZERO=I
              PARRAY(I)=QI(J*DX)
          100 CONTINUE
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:41:45    14-Dec-87           Page 19
CPFSKFFT.FTN;23         /F77/TR:BLOCKS/WR

              DOUBLE PRECISION FUNCTION QI(X)
      C
      C THE FUNCTION QI(X)
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              EXTERNAL DGAU20, QINNER
              PARAMETER (HALFPI = 1.5707963267948966192313200)
              PARAMETER (HALMPI =-1.5707963267948966192313200)
              PARAMETER (TWOPI=6.2831853071795864769253600)
              DIMENSION WORK(15), STACK(15), HEAP(15)
              COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
              COMMON /QPAR/ UQ, VQ, WQ, TERM
              COMMON /ECKS/ WCPMX, RCX
              WCPMX=WQ*COSINE(DELPHI-X)
              RCX=ARE*COSINE(X)
              CALL ADQUAD(HALMPI, HALFPI, RESULT, DGAU20, QINNER, 1.D-6,
          $               1.D-6, WORK, STACK, HEAP, 15, KODE)
              IF(KODE.NE.0) STOP 'QI INTEGRAL FAILED'
              QI=OMRSQ2*RESULT/TWOPI
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:41:48    14-Dec-87           Page 20
CPFSKFFT.FTN;23         /F77/TR:BLOCKS/WR

              DOUBLE PRECISION FUNCTION QINNER(U)
      C
      C THE INTEGRAND FUNCTION FOR QI
      C
              IMPLICIT DOUBLE PRECISION(A-H,O-Z)
              COMMON /QPAR/ UQ, VQ, WQ, TERM
              COMMON /ECKS/ WCPMX, RCX
              CU=COSINE(U)
              SU=DSIN(U)
              AAA=(UQ-VQ*SU-WCPMX*CU)/(1.DO-RCX*CU)
              QINNER=CU*DEXP(-AAA)*(TERM-AAA)
              RETURN
              END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:41:50    14-Dec-87           Page 21
CPFSKFFT.FTN;23         /F77/TR:BLOCKS/WR

              SUBROUTINE BUILD(CU, CJ, PU, NOU, PJ, NOJ, Z, K, NR, CS)
      C
      C BUILD A PDF PREPARATORY TO TAKING FFT
      C
              IMPLICIT DOUBLE PRECISION (A-H,O-Z)
              PARAMETER (IDIM=8192)
              PARAMETER (NPI=128)
              PARAMETER (N2PI=2*NPI)
              VIRTUAL Z(2,IDIM)
              DIMENSION CU(4), CJ(4), PU(256), PJ(256)
              COMMON /PARMS/ EBNODB, EBNJDB, GAMMA, L, H, D
              COMMON /SIZES/ ISIZE, JSIZE, LSIZE, ISIZE2
              OMG=1.DO-GAMMA
              IF(NOU.NE.NOJ) STOP 'DELPHI DEPENDS ON JAMMING'
      C CONSTRUCT MAIN LOBE (NO CLICKS)
              GP=GAMMA*CJ(1)
              OGP=OMG*CU(1)
              CS=GP+OGP
              DO 10 I=1,256
              ISUB=I-NOU+1
              IF(ISUB.GT.0) NR=ISUB
              IF(ISUB.LE.0) ISUB=ISUB+ISIZE
              Z(1,ISUB)=OGP*PU(I) + GP*PJ(I)
              Z(2,ISUB)=0.DO
         10   CONTINUE
              IF(L.EQ.1) RETURN
      C SHIFTED LOBES (WITH CLICKS)
              DO 100 M=1,K
              N2KPI=M*N2PI
              CP=GAMMA*CJ(M+1)
              OGP=OMG*CU(M+1)
              CS=CS+GP+OGP
              DO 50 I=1,256
              ISUB=I-NOU+1-N2KPI
              IF(ISUB.LE.0) ISUB=ISUB+ISIZE
              Z(1,ISUB)=OGP*PU(I) + GP*PJ(I)
              Z(2,ISUB)=0.DO
         50   CONTINUE
        100   CONTINUE
              RETURN
              END
```

```
0001            SUBROUTINE PSUBE(Z, NRIGHT, PE)
        C
        C COMPUTE THE PROBABILITY OF ERROR BY SIMPSON'S RULE INTEGRATION
        C
0002            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
0003            PARAMETER (IDIM=8192)
0004            PARAMETER (PI=3.14159265358979323846D0)
0005            VIRTUAL Z(2,IDIM)
0006            COMMON /SIZES/ ISIZE, JSIZE, LSIZE, ISIZE2
0007            COMMON /PARMS/ EBNODB, EBNJDB, GAMMA, L, H, D
0008            COMMON /THINGS/ ARE, DELPHI, OMRSQ2, DX, NPHI
0009            IMAX=L*NRIGHT+1
0010            IF(MOD(IMAX,2).EQ.0) THEN
        C HAVE EVEN NUMBER OF POINTS, DROP BACK ONE AND FINISH LAST LITTLE
        C PIECE BY TRAPEZOIDAL RULE
0011              IMAX=IMAX-1
0012              ADD=0.5D0*DX*(Z(1,IMAX)+Z(1,IMAX+1))
0013            ELSE
0014              ADD=0.D0
0015            END IF
        C
        C DO MAIN INTEGRAL BY SIMPSON'S RULE
        C
0016            SIMP=Z(1,1)+Z(1,IMAX)
0017            WT=2.D0
0018            DO 10 I=2,IMAX-1
0019            WT=6.D0-WT
0020            SIMP=SIMP+WT*Z(1,I)
0021       10   CONTINUE
0022            SIMP=SIMP*DX/3.D0+ADD
        C
        C APPLY NORMALIZATION OMITTED FROM INVERSE FFT
        C
0023            SIMP=SIMP*DXI(DX,L-1)/ISIZE
        C
        C ERROR PROBABILITY = 1 - CORRECT PROBABILITY
        C
0024            PE=1.D0-SIMP
0025            RETURN
0026            END
```



```
0001            SUBROUTINE DVFFTS(NPTS, NN2, X, DIR, C, IVALID)
        C
        C FAST FOURIER TRANSFORM OF COMPLEX DATA ARRAY; OUTPUT SCRAMBLED ORDER
        C
        C ROBERT H. FRENCH   9 MARCH 1983, MODIFIED 16 SEPT. 1987
        C
        C ADAPTED FROM PROGRAMS IN JAMES R. FISHER, "FORTRAN PROGRAM
        C FOR FAST FOURIER TRANSFORM," NRL REPORT 7041, NAVAL RESEARCH
        C LABORATORY, WASHINGTON, D.C., APRIL 16, 1970.
        C
        C
        C   NPTS - LENGTH OF DATA ARRAY
        C    NN2 - SIZE OF WORK ARRAY, N2 >= NPTS/4+1
        C      X - DATA ARRAY, COMPLEX, LENGTH NPTS
        C    DIR - DIRECTION: +1.0 FOR FFT, -1.0 FOR INVERSE FFT
        C      C - WORK ARRAY FOR COEFFICIENTS, LENGTH NPTS/4+1
        C IVALID - LOGICAL VARIABLE, SET .TRUE. IF C AND IWORK HOLD VALID
        C          DATA AS A RESULT OF A PREVIOUS CALL TO FFT
        C
0002            IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003            DIMENSION E(2)
0004            VIRTUAL X(2,NPTS), C(NN2)
0005            LOGICAL IVALID
        C
        C ----------------------------------------------------------
        C
0006            ISUB=NPTS/4
0007            N2=NPTS/2
0008            IF(.NOT.IVALID) THEN
        C
        C COMPUTE SINE/COSINE TABLE
        C
0009              ANG=3.14159265358979323846D0/(NPTS/2)
0010              C(1)=1.D0
0011              C(ISUB+1)=0.D0
        C
0012              DO 1 J=1,ISUB-1
0013              C(J+1)=DCOS(J*ANG)
0014        1     CONTINUE
0015              IVALID=.TRUE.
0016            END IF
        C
        C DO THE FFT
        C
0017            INCT=1
0018            LINC=NPTS/2
0019       10   LOC=LINC
0020       40   ITHTA=0
        C  80   E=CMPLX(C(ITHTA+1),-DIR*C(ISUB+1-ITHTA))
0021       80   E(1)=C(ITHTA+1)
0022            E(2)=-DIR*C(ISUB+1-ITHTA)
0023       60   LOC=LOC+1
```

```
            LOC1=LOC-LINC
      C     E=E*(X(LOC1)-X(LOC))
            W1=X(1,LOC)
            W2=X(2,LOC)
            Y1=X(1,LOC1)
            Y2=X(2,LOC1)
            TEMP=E(1)*(Y1-W1)-E(2)*(Y2-W2)
            E(2)=E(1)*(Y2-W2)+E(2)*(Y1-W1)
            E(1)=TEMP
      C     X(LOC1)=X(LOC1)+X(LOC)
            X(1,LOC1)=Y1+W1
            X(2,LOC1)=Y2+W2
      C     X(LOC)=E
            X(1,LOC)=E(1)
            X(2,LOC)=E(2)
            ITHTA=ITHTA+INCT
            IF(ITHTA-N2)20,30,30
      C
         20 IF(ITHTA-ISUB)80,50,50
      C
      C  50 E=CMPLX(-C(N2+1-ITHTA),-DIR*C(ITHTA-ISUB+1))
         50 E(1)=-C(N2+1-ITHTA)
            E(2)=-DIR*C(ITHTA-ISUB+1)
            GOTO 60
      C
         30 IF(LOC-NPTS)90,91,91
      C
         90 LOC=LOC+LINC
            GOTO 40
      C
         91 IF(2-LINC) 92,93,94
      C
         92 LINC=LINC/2
            INCT=INCT+INCT
            GOTO 10
      C
         93 DO 100 LOC=2,NPTS,2
      C     E=X(LOC-1)-X(LOC)
            E(1)=X(1,LOC-1)-X(1,LOC)
            E(2)=X(2,LOC-1)-X(2,LOC)
      C     X(LOC-1)=X(LOC-1)+X(LOC)
            X(1,LOC-1)=X(1,LOC-1)+X(1,LOC)
            X(2,LOC-1)=X(2,LOC-1)+X(2,LOC)
      C     X(LOC)=E
            X(1,LOC)=E(1)
            X(2,LOC)=E(2)
        100 CONTINUE
         94 RETURN
            END
```



```
            SUBROUTINE DISFFT(X, N, NN2, L2N, C)
      C
      C DOUBLE PRECISION INVERSE FFT OF SCRAMBLED INPUT ARRAY
      C
      C    X = SCRAMBLED ORDER INPUT ARRAY
      C    C = COS/SIN ARRAY (MUST BE PRE-COMPUTED)
      C    N = LENGTH OF X
      C  NN2 = N/4+1 = LENGTH OF C
      C  L2N = LOG(BASE 2) OF N
      C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            VIRTUAL X(2,N), C(NN2)
            K=1
            N2=N/2
            N4=N/4
            N34=3*N4
            DO 5 L=1,L2N
            M=K
            BM=M
            K=2*K
            W1=1.D0
            W2=0.D0
            DO 5 J=1,M
            BJ=J
            DO 4 I=J,N,K
            I2=I+M
            B1=X(1,I2)*W1-X(2,I2)*W2
            B2=X(1,I2)*W2+X(2,I2)*W1
            T1=X(1,I)-B1
            T2=X(2,I)-B2
            X(1,I)=X(1,I)+B1
            X(2,I)=X(2,I)+B2
            X(1,I2)=T1
            X(2,I2)=T2
          4 CONTINUE
            ISUB=J*(N2/M)
            IF(ISUB.LE.N4) THEN
              W1=C(ISUB+1)
              W2=C(N4+1-ISUB)
            ELSE IF(ISUB.LE.N2) THEN
              W1=-C(N2+1-ISUB)
              W2=C(ISUB-N4+1)
            ELSE IF(ISUB.LE.N34) THEN
              W1=-C(ISUB-N1+1)
              W2=-C(N34+1-ISUB)
            ELSE
              W1=C(N+1-ISUB)
              W2=-C(ISUB-N34+1)
            END IF
          5 CONTINUE
            RETURN
            END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:42:11     14-Dec-87          Page 26
CPFSKFFT.FTN;23          /F77/TR:BLOCKS/WR

              SUBROUTINE ADQUAD(XL,XU,Y,QR,F,TOL,ABSTOL,
        $                       WORK,STACK,HEAP,N,KODE)
      C
      C ADAPTIVE QUADRATURE ALGORITHM FOR NUMERICAL INTEGRATION
      C
      C     XL - LOWER LIMIT OF INTEGRAL (IN)
      C     XU - UPPER LIMIT OF INTEGRAL (IN)
      C      Y - VALUE OF INTEGRAL (OUT)
      C     QR - NAME OF A QUADRATURE RULE SUBROUTINE (IN)
      C               WITH CALLING SEQUENCE
      C               CALL QR(XL,XU,F,Y)
      C      F - NAME OF FUNCTION TO BE INTEGRATED (IN)
      C    TOL - ERROR TOLERANCE FOR FINAL ANSWER (IN)
      C ABSTOL - ABSOLUTE ERROR TOLERANCE
      C   WORK - WORK ARRAY OF SIZE N (IN)
      C  STACK - SECOND WORK ARRAY OF SIZE N, MUST NOT BE
      C   HEAP - THIRD WORK ARRAY, SIZE N, DISTINCT FROM WORK AND STACK
      C          SAME ARRAY AS WORK (IN)
      C      N - SIZE OF WORK AND STACK; MAX. NO. OF BISECTIONS (IN)
      C   KODE - ERROR INDICATOR (OUT)
      C          0 — NO ERROR
      C          1 — WORK ARRAYS TOO SMALL
      C
      C R. H. FRENCH, 14 AUGUST 1984
      C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            EXTERNAL F
            DIMENSION WORK(N),STACK(N),HEAP(N)
            KODE=0
            Y=0.D0
            WORK(1)=XU
            CALL QR(XL,XU,F,T)
            HEAP(1)=T
            A=XL
            NPTS=1
            EPS=TOL
            STACK(1)=EPS
      10    B=WORK(NPTS)
            XM=(A+B)*0.5D0
            CALL QR(A,XM,F,P1)
            CALL QR(XM,B,F,P2)
            TEST=DMAX1(EPS*DABS(T),ABSTOL)
            IF(DABS(T-P1-P2).LE.TEST .OR. DABS(T).LE.ABSTOL) GOTO 20
      C SPLIT IT
            NPTS=NPTS+1
            IF(NPTS.GT.N) THEN
              Y=P1+P2
              KODE=1
              RETURN
            END IF
            WORK(NPTS)=XM
            HEAP(NPTS)=P2
```

```
PDP-11 FORTRAN-77 V4.0-1        07:42:11     14-Dec-87          Page 27
CPFSKFFT.FTN;23          /F77/TR:BLOCKS/WR

            T=P1
            EPS=DMAX1(EPS/2.D0,5.D-16)
            STACK(NPTS)=EPS
            GOTO 10
      C FINISHED A PIECE
      20    Y=Y+P1+P2
            EPS=STACK(NPTS)
            T=HEAP(NPTS)
            NPTS=NPTS-1
            A=B
            IF(NPTS.EQ.0) RETURN
            GOTO 10
            END
```

```
PDP-11 FORTRAN-77 V4.0-1        07:42:15     14-Dec-87          Page 28
CPFSKFFT.FTN;23          /F77/TR:BLOCKS/WR

            DOUBLE PRECISION FUNCTION COSINE(X)
      C
      C INSURE DCOS(0.D0) = 1.0000000000000000D0 EXACTLY
      C
            IMPLICIT DOUBLE PRECISION(A-H,O-Z)
            IF(X.EQ.0.D0) THEN
              COSINE=1.D0
            ELSE
              COSINE=DCOS(X)
            END IF
            RETURN
            END
```

```
0001          SUBROUTINE DGAU20(A,B,F,ANSWER)
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        C                                                     C
        C 20-POINT GAUSSIAN INTEGRATION OVER ARBITRARY INTERVAL C
        C                                                     C
        C REF.: ABRAMOWITZ & STEGUN, EQ. 25.4.30 AND TABLE 25.4 C
        C                                                     C
        C R. H. FRENCH, 21 JUNE 1983                          C
        C                                                     C
        CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
0003          DIMENSION X(10),W(10)
0004          DATA X/ 0.07652652113349733375D0,
            1         0.22778585114164507808D0,
            1         0.37370608871541956067D0,
            1         0.51086700195082709800D0,
            1         0.63605368072651502545D0,
            1         0.74633190646015079261D0,
            1         0.83911697182221882339D0,
            1         0.91223442825132590586D0,
            1         0.96397192727791379126D0,
            1         0.99312859918509492478D0 /
0005          DATA W/ 0.15275338713072585069D0,
            1         0.14917298647260374678D0,
            1         0.14209610931838205132D0,
            1         0.13168863844917662689D0,
            1         0.11819453196151841731D0,
            1         0.10193011981724043503D0,
            1         0.08327674157670474872D0,
            1         0.06267204833410906357D0,
            1         0.04060142980038694133D0,
            1         0.01761400713915211831D0 /
0006          ANSWER=0.D0
0007          BMAO2=(B-A)/2.D0
0008          BPAO2=(B+A)/2.D0
0009          DO 10 I=1,10
0010          C=X(I)*BMAO2
0011          Y1=BPAO2+C
0012          Y2=BPAO2-C
0013          ANSWER=ANSWER+W(I)*(F(Y1)+F(Y2))
0014    10    CONTINUE
0015          ANSWER=ANSWER*BMAO2
0016          RETURN
0017          END
```